slide1:
Good afternoon, everyone. Today, we begin Lecture 10, which focuses on networks. Before we dive into the details, let me remind you that this topic builds directly on what we've covered in linear systems and Fourier analysis. If you haven't reviewed those concepts recently, now would be a great time to revisit them, because they'll help you see the deeper connections between the theory and the applications we'll explore today.

slide2:
We are right on schedule in our journey through this material. So let's start.

slide3:
Before we dive into networks, let's recall the two related ideas we've already discussed — functions and systems.
A function is a mathematical concept. It takes an input, processes it according to some rule, and produces an output. You can think of it in terms of a domain and a range.#A system is essentially the same idea, but we often use this term in engineering. We talk about inputs and outputs, and the system could be simple or complex.
Now, a network is really just a more complex kind of system — one made up of many interconnected subsystems. The internet is a perfect example: millions of computers linked together, exchanging information. From a mathematical perspective, you can think of a network as a very complicated function, often involving many variables and composite functions linked together.
In electrical engineering, we often study electrical networks, where components like resistors, capacitors, and inductors are connected to form a circuit. In modern artificial intelligence, we have neural networks, which connect many artificial neurons to process information — in some ways, similar to how biological neurons communicate through electrical signals in our nervous system.
Today, we'll start with the more classic engineering example: the electrical network.

slide4:
Let's start with the most basic electrical component — the resistor.
You already know Ohm's Law, which tells us that the voltage across a resistor equals the current flowing through it multiplied by its resistance. In words,#v equals i times R.
We can also rearrange this to say that the current i equals the voltage v divided by the resistance R.#This means the current is directly proportional to the applied voltage and inversely proportional to the resistance.
The name resistor comes from its function — it resists the flow of electric current. In other words, it limits how many electrons can pass through it for a given voltage.
Next, we'll move on to another fundamental component — the capacitor.

slide5:
Here we have the capacitor — another fundamental circuit component.
When we connect a voltage source, such as a battery, across the capacitor, electrons are pushed toward one plate, creating a buildup of negative charge. On the opposite plate, a positive charge accumulates. Importantly, the capacitor itself doesn't let electrons pass directly through; instead, the charges stay separated by an insulating material called the dielectric.
As more charge accumulates, an opposing electric field develops across the plates. Eventually, this field balances the applied voltage, and the current flow stops.
Mathematically, the relationship between voltage and current for a capacitor is given by:#I equals C times the derivative of v with respect to time.#In words, the current is proportional to how quickly the voltage changes over time.#We can also write it in integral form:#v equals one over C times the integral of i with respect to time, plus the initial voltage.
This is different from a resistor, where voltage and current are simply proportional. For a capacitor, the key is the rate of voltage change — no change in voltage means no current flows.

slide6:
Now, let’s look at the inductor. Its voltage–current relationship is similar in form to the capacitor’s, but with the roles of voltage and current reversed.
For an inductor, the voltage v is equal to the inductance L multiplied by the rate of change of current — in other words,#v equals L times the derivative of i with respect to time.
We can also express this in integral form:#I equals one over L times the integral of v with respect to time, plus the initial current.
So, for a capacitor, current depends on how quickly voltage changes. For an inductor, voltage depends on how quickly current changes. Capacitors store energy in an electric field, while inductors store energy in a magnetic field.
These three components — resistor, capacitor, and inductor — are the building blocks of classic electrical circuits. Understanding their voltage–current relationships is essential before we move on to phasors, which will make analyzing AC circuits much easier.

slide7:
Let’s take a moment to revisit complex numbers, because they form the foundation for understanding phasors.
A complex number has two parts — a real part, x, and an imaginary part, y. We can represent this on a complex plane, with the horizontal axis for the real part and the vertical axis for the imaginary part. This is like a Cartesian coordinate system.
But there’s another way to represent the same point — using polar coordinates. Instead of x y, we describe it by its distance from the origin, called the magnitude r, and its angle from the real axis, called the phase angle phi.
Mathematically, we can write this as r times e to the power i phi. Here, e to the power i phi comes from Euler’s formula, which says e to the i phi equals cosine phi plus i times sine phi. Multiplying r by those components gives us back x y.
When we analyze sinusoidal signals, like in AC circuits, we often know the frequency in advance. In that case, the important quantities are the magnitude r, which tells us the signal’s strength, and the phase angle phi, which tells us where the waveform starts in time.
Instead of writing out the full exponential, we can use the simpler phasor notation: r angle phi. For example, if we know an AC current has a magnitude of 5 amperes and a phase angle of 30 degrees, we can write it as “5 at angle 30 degrees.”
This phasor approach is a very convenient way to describe sinusoidal signals, and as we’ll see, it makes circuit analysis much simpler.

slide8:
Let’s take a closer look at the imaginary unit i and understand it from a geometric perspective using Euler’s formula:#e to the i theta equals cosine theta plus i times sine theta.
If we set theta to 90 degrees, e to the i 90° gives us i. You can think of multiplying by i as rotating a vector by 90 degrees in the complex plane. For example, if we start with the unit vector at 1 on the real axis and multiply by i, it rotates to point straight up along the imaginary axis.
If we multiply by i again — another 90-degree rotation — we end up at negative 1 on the real axis. In other words, i times i equals –1. That’s the origin of the idea that i is the square root of negative one.
More generally, multiplying by e to the i theta rotates a vector by an angle theta, while multiplying by a real number r simply scales its length. This gives a very natural, visual way to think about complex multiplication — scaling changes the magnitude, and the exponential term changes the direction.
Addition is also straightforward: if you have two complex numbers, u and v, you add their real parts and their imaginary parts separately. Graphically, that’s just adding two vectors tip-to-tail to get u plus v.
With these rules for addition and multiplication, we have a complete algebraic system for complex numbers. And this system is extremely powerful for representing sinusoidal waves — the building blocks of Fourier analysis — as well as for solving wave equations in fields like ultrasound and electromagnetics.

This is why the phase representation, using complex numbers, is such a fundamental tool in engineering and physics.

slide9:
Now let’s see how phase, or phasor, representation applies directly to a capacitor’s voltage–current relationship.
In basic DC circuits, Ohm’s Law tells us that voltage equals current times resistance. But for a capacitor, we normally describe the relationship in terms of a derivative or an integral — for example, i equals C times dv over dt. However, when we work in the phasor domain, something elegant happens. The derivative operation in the time domain becomes a simple multiplication by j omega in the frequency domain. This means that, for sinusoidal signals, the voltage–current relationship for a capacitor can look just like Ohm’s Law — except that instead of a real resistance, we have a complex quantity called impedance.
Here’s the key point: in any linear circuit made up of resistors, capacitors, and inductors, if we drive it with a pure sinusoidal signal, the output will also be a sinusoid at the exact same frequency. The amplitude may change, and the waveform may shift in phase, but the frequency remains unchanged.
For example, suppose the voltage across a capacitor is A cosine omega t. Differentiating this to find the current shifts the waveform by 90 degrees and multiplies its amplitude by C omega. In phasor notation, this phase shift is written as “at an angle of + 90 degrees.”
Finally, in the phasor domain, the capacitor’s impedance is 1 over j omega C. That’s directly analogous to resistance in Ohm’s Law — except here it’s a complex value, reflecting the fact that voltage and current are out of phase.

slide10:
For an inductor, the process is very similar to what we saw with a capacitor, but here it’s the current that we start with.
Suppose the current is sinusoidal: B cosine omega t. The voltage across the inductor is given by v equals L times di/dt. Differentiating the current shifts the waveform by 90 degrees and changes the amplitude by a factor of L omega.
In phasor notation, if the current is B at angle 0 degrees, then the voltage becomes L omega B at angle 90 degrees. That phase shift means the voltage leads the current by 90 degrees in an inductor — the exact opposite of a capacitor, where the current leads the voltage by 90 degrees.
In the phasor domain, this relationship is written simply as V equals j omega L times I. Here, j omega L plays the same role for an inductor that resistance R plays for a resistor — it’s the impedance. The only difference is that it’s a complex quantity, reflecting the phase relationship between voltage and current.
So once again, phasor notation allows us to generalize Ohm’s Law to all three basic components — resistor, capacitor, and inductor — using the single idea of impedance.

slide11:
Up to now, we’ve talked about resistors, capacitors, and inductors separately. Each has its own voltage–current relationship, and each can be expressed in the phasor domain.
Instead of calling all these effects “resistance,” we use a broader term — impedance. Impedance describes how any circuit element opposes the flow of alternating current, and it applies to all three components.
For a resistor, impedance is simply R.#For a capacitor, impedance is 1 over j omega C.#For an inductor, impedance is j omega L.
In each case, impedance is defined as voltage divided by current — Z equals V over I. This is the same structure as Ohm’s Law, but with Z taking the place of R.
When we combine components in a circuit, the total impedance depends on whether they are connected in series or in parallel. Later, we’ll learn how to calculate equivalent impedance for these different configurations. But the important takeaway here is that impedance unifies the behavior of resistors, capacitors, and inductors into one general concept, making AC circuit analysis much simpler.

slide12:

Let’s now step back and look at a general electrical network. I’ll use this simple example to introduce three important terms: node, branch, and loop.
A node is a point where two or more circuit components meet. These components could be resistors, capacitors, inductors, or sources like batteries and current generators. In our diagram, each place where elements connect is a node.
A branch is a single circuit element between two nodes. You can think of it as one “arm” of the circuit, carrying current from one node to another.
A loop is a closed path that goes through a sequence of branches and nodes without passing through any branch or node more than once. In other words, it’s a single, non-repeating path that starts and ends at the same point.
Using these definitions, you might ask: in this diagram, how many nodes are there? And how many loops? This kind of reasoning will become important when we apply Kirchhoff’s laws for circuit analysis.

slide13:
So, how many nodes are in this circuit? Let’s apply the definition.
A node is any point where two or more branches connect. Here, all the points along this conducting path are electrically the same — the wires are ideal conductors with zero resistance — so we treat them as one single node. This first node connects resistor R 1, resistor R 2, resistor R 3, and the voltage source.
The other connection point, down here, is the second node. It ties together the lower ends of R 2 and R 3, the negative terminal of the voltage source, and the current source.
So even though there are multiple connection spots physically drawn, electrically, we have just two distinct nodes in this circuit.

slide14:
Now let’s talk about loops — and more specifically, independent loops.
A loop is a closed path in the circuit that passes through a sequence of branches and returns to the starting point without crossing any branch more than once. But not all loops are equally useful in analysis. We focus on independent loops, which each contain at least one branch that is not part of any other loop in the set.
In this example, we have three independent loops:
Loop 1 includes the voltage source, resistor R 1, and resistor R 2.
Loop 2 includes resistor R 2 and resistor R 3.
Loop 3 includes resistor R 3 and the current source.
You might notice that other closed paths exist — for example, one that goes all the way around the outside — but these can be derived from combinations of the independent loops. They don’t give us any new information.
The reason this concept is important is that each independent loop gives us one equation when we apply Kirchhoff’s Voltage Law. These equations are the building blocks for solving circuit problems with multiple unknowns, such as the current through each branch or the voltage across each resistor.

slide15:
The first rule we use in circuit analysis is Kirchhoff’s Current Law, or KCL. It’s very straightforward: for any node in a circuit, the total current flowing into the node must equal the total current flowing out.
Think of the node as a junction — electrons flow in from some branches and out through others. They don’t pile up or vanish at the node. If two amperes flow in from one branch and three amperes flow in from another, then a total of five amperes must flow out through the remaining branch.
You can picture this like people moving through a doorway: the number of people entering must match the number leaving, assuming no one stays inside the doorway. Or like cars passing through an intersection: cars can’t just appear or disappear; what comes in must go out.
KCL is essentially a statement of conservation of charge — charge is neither created nor destroyed at a node. This simple but powerful rule will be one of the main tools we use to solve circuit equations.

slide16:
The second key rule is Kirchhoff’s Voltage Law, or KVL. It states that if you

travel around any closed loop in a circuit, the sum of all the voltage rises and drops must equal zero.
You can think of it like taking a hike around a loop trail that starts and ends at the same place. Along the way, you might climb uphill — that's like a voltage rise from a battery — and you might walk downhill — that's like a voltage drop across a resistor. By the time you return to where you started, your total elevation change is zero.
In a circuit, it's the same idea but with electrical potential instead of gravitational potential. As electrons move through a voltage source, they gain energy; as they pass through resistive elements, they lose energy. Around a complete loop, the total gains and losses cancel out exactly — a direct consequence of energy conservation.
In this example, the voltage source V 1 is 10 volts. As we move around the loop, we see drops of 2 volts, 3 volts, and 1 volt across different resistors, and zero volts across the ideal conducting wire. That leaves one resistor with an unknown drop. To satisfy KVL, it must drop 4 volts so that the algebraic sum is zero.
When we combine KVL with Kirchhoff's Current Law, we can write enough equations to solve for all the unknown currents and voltages in even a complex electrical network.

slide17:
If you start with the simplest case — a single loop — the problem is straightforward. You can solve it directly, just like using Ohm's Law: voltage equals impedance times current.
As you make the network more complex, you can add loops, nodes, or both. Each time you add a loop, you introduce a new unknown, but KVL gives you one more independent equation to match it. Each time you add a node, you also introduce a new unknown; however, KCL provides the additional equation needed.
If you add both a node and a loop at the same time, you introduce two unknowns — for example, splitting a branch into two paths so currents can differ — but you also gain two equations: one from KCL at the new node and one from KVL around the new loop.
The key point is that no matter how you expand the network, KCL and KVL together give you exactly the number of independent equations you need to solve for all the unknown currents and voltages. This balance is what ensures that a well-defined electrical network always has a unique solution.

slide18:
Let's solve this example step by step.
We have two voltage sources and two loops. Our goal is to find three unknown currents:
I 1, which flows through the left loop,
I 2, which flows through the right loop,
and I_3, which flows along the bottom branch.
First, we apply Kirchhoff's Current Law at node c. In words, the current I 1 flowing into the node plus the current I 2 flowing into the node must equal the current I_3 flowing out. That's our first equation:#"I one plus I two equals I three."
Next, we apply Kirchhoff's Voltage Law to the green loop, moving clockwise through points b, e, f, c, and back to b. As we go around the loop:
passing through the fourteen-volt battery in the opposite direction gives a minus fourteen volts,
passing through the ten-volt battery in the opposite direction gives minus ten volts,
passing through the six-ohm resistor with current I 1 gives plus six times I 1,
and passing through the four-ohm resistor with current I 2 gives minus four times I 2.
Adding them all up and setting the total to zero, we have:#"negative fourteen, minus ten, plus six times I one, minus four times I two, equals zero."
Finally, we apply Kirchhoff's Voltage Law to the blue loop, moving through points a, b, c, d, and back to a. This gives:#"ten, minus six times I one, minus two times I three, equals zero."
Now we have three equations:

I one plus I two equals I three.
Negative fourteen minus ten plus six I one minus four I two equals zero.
Ten minus six I one minus two I three equals zero.
Solving these, we find:#I one equals two amperes,#I two equals negative three amperes — meaning it actually flows in the opposite direction we assumed —#and I three equals negative one ampere.

slide19:
This slide is a summary that generalizes Ohm's Law for alternating current circuits.
In direct current circuits, we know the familiar form: "voltage equals resistance times current." In alternating current circuits, thanks to phasor notation, we can keep the same simple structure — voltage equals impedance times current — but now impedance can be a complex number, depending on the type of component.
For a resistor, impedance is simply R, and voltage and current are in phase — their peaks occur simultaneously.
For a capacitor, the impedance is "one divided by j omega C," which in polar form is "one over omega C at an angle of negative ninety degrees." This means the voltage lags the current by ninety degrees. In other words, the current reaches its peak before the voltage does.
For an inductor, the impedance is "j omega L," or in polar form "omega L at an angle of positive ninety degrees." This means the voltage leads the current by ninety degrees — the voltage peaks first, and the current builds up afterward.
If you think about it intuitively:
An inductor resists sudden changes in current, so when you first apply voltage, the current starts small and gradually increases.
A capacitor resists changes in voltage, so when you first apply current, the voltage takes time to build.
Mathematically, we derive these results from the voltage–current relationships we saw earlier, but phasor notation makes them look just like Ohm's Law — only with complex impedance replacing simple resistance.

slide20:
Let's learn how to replace two impedances with a single equivalent impedance.
Series connection — one after the other on the same branch:#The same current flows through both, and the voltage drops add.
So the equivalent is:#"Z sub e equals Z one plus Z two."#This works even when the impedances are complex.
A one-line proof, spoken: assume a current I flows.#The total voltage is I times Z sub e.#It is also I times Z one plus I times Z two.#Set them equal and cancel I: "Z sub e equals Z one plus Z two."
Parallel connection — the two elements share the same two nodes:#The voltage across each is the same, and the currents add.
So the equivalent is:#"one over Z sub e equals one over Z one plus one over Z two,"#or, equivalently,#"Z sub e equals Z one times Z two, divided by Z one plus Z two."
Quick proof, spoken: let the common voltage be V.#Currents are I 1 = V over Z 1 and I 2 = V over Z 2.#Total current I = I 1 plus I 2 = V times (1 over Z 1 plus 1 over Z 2).#But I = V over Z sub e.#Cancel V to get the reciprocal form above.

slide21:
Once we know how to combine impedances into a single equivalent value, we can simplify any branch of a network. Each branch can then be treated as just two things: an impedance and a source — either a voltage source, a current source, or both. If there are more components in the branch, we simply combine them into one equivalent impedance before solving.
After that, solving the network becomes a two-step process:#Step one: find the overall currents and voltages using the simplified equivalent impedances.#Step two, if needed, break the equivalent impedance back into its original components and distribute the results to each part.
Now let's apply this to the idea of a voltage divider. In the circuit on the left, we have an input voltage split between a resistor and a capacitor. The fraction of the input voltage that appears across the capacitor is given by the

transfer function:
“Capital T of omega equals one divided by j omega C, all over R plus one divided by j omega C.”
If we simplify, that becomes:#“T of omega equals one over one plus j times omega over omega sub c,”#where omega sub c equals one over R times C.
This formula tells us how the circuit responds to different frequencies. If the input is a pure sinusoidal wave, we just plug its frequency into omega and get the scaling factor for the output.
And if the input is an arbitrary waveform, Fourier analysis lets us break it into many sinusoidal components, each with its own frequency.
We apply the transfer function to each frequency separately, then sum them back up to get the total output.
This particular circuit is called a low-pass filter because it passes low-frequency signals with little attenuation, but strongly reduces high-frequency components. At very low frequencies, omega is close to zero, so the transfer function is close to one — the output nearly matches the input. At very high frequencies, omega is large, the transfer function becomes very small, and the output is greatly reduced.
The inductor–resistor version on the right works the same way, except its cutoff frequency omega sub c equals R divided by L.

slide22:
When we move into the Fourier domain, any signal can be represented as a collection of frequency components. Filters are circuits or systems that selectively pass or block these components based on their frequency.
A low-pass filter allows low-frequency components to pass through while attenuating or reducing high-frequency components. In the idealized diagram here, the passband is shown as a flat rectangle up to the cutoff frequency, omega sub c, and then it drops to zero in the stopband. In real circuits, the transition is not perfectly sharp — it’s more gradual — but the concept is the same. At the cutoff frequency, the signal power has dropped to half its low-frequency value.
A high-pass filter does the opposite: it blocks low-frequency signals and passes the high-frequency ones.
A band-pass filter only allows frequencies within a certain range to pass, blocking both lower and higher frequencies.
A band-stop filter, sometimes called a notch filter, removes frequencies within a specific range but passes those outside that range.
All of these can be analyzed using the same approach:
Decompose the input signal into sinusoidal components using Fourier analysis.
For each frequency, calculate the system’s response using impedance and the phasor form of Ohm’s Law: voltage equals impedance times current.
Combine the individual frequency responses to reconstruct the total output.
For DC circuits with constant voltages, we only need algebraic equations — just resistance times current equals voltage. But with AC signals or sinusoidal sources, the relationships involve derivatives and integrals. Phasor notation transforms those into simple algebraic equations, letting us solve AC problems with the same ease as DC problems.

slide23:
In this sense, an alternating current circuit can be treated just like a direct current circuit — once we use phasor notation and impedance. That’s why we say AC is DC in disguise.
By representing AC voltages and currents as phasors, we turn differential equations into simple algebraic ones, just like in DC analysis. This dramatically simplifies the process of solving AC circuit problems.
This clever idea was first developed decades ago by an engineer at the GE Global Research Center, not far from here. It’s now a standard computational tool in electrical engineering, and it’s one of the reasons AC circuit analysis can be so elegant.

slide24:
Here’s a very typical and practical circuit — the voltage divider.
We start with an input voltage, which is split between two resistors or

impedances, R 1 and R 2. The voltage drop across the second resistor becomes our output signal. This is a common way to pass a signal from one stage of a system to the next.
A real-world example might be a sensor connected to a measurement device. The human body, for instance, has its own electrical impedance. If we were measuring a cardiac signal, we could model the body's impedance as R 1. The second resistor, R 2, could represent the input impedance of our measuring instrument.
From an engineering perspective, we usually have three goals when designing such a circuit:
Large input resistance — We want R1 to be large so that most of the input voltage appears across it, ensuring we capture as much of the signal as possible.
Small output resistance — We want R2 to be small so the next stage in the circuit can take the signal without losing much voltage.
High gain — If possible, we want to amplify the signal to make it easier to detect and process.
These goals can sometimes conflict — for example, making R 1 large helps with capturing the signal, but we also want R 2 small for driving the next stage. Circuit design is about balancing these trade-offs.
In the example at the bottom right, R S is our sensor, whose resistance changes slightly by a factor of x in response to something like temperature or pressure. R_L is the load resistor. If x equals zero, the sensor is at its nominal value. When x changes, even by a small amount, the output voltage V out changes.
Mathematically, V out equals V in multiplied by the ratio R 2 over R 1 plus R 2. This is just the voltage divider rule we've seen before. In practice, the change in x might be very small, so the change in V out is tiny — often just a few millivolts. If we want to amplify this tiny change, we ideally remove the constant background voltage so that only the small variation gets amplified. This allows for much higher gain without overloading the system.

slide25:
One way to remove the background signal and make small changes easier to detect is by using a Wheatstone bridge.
The Wheatstone bridge is essentially two voltage dividers connected side by side. On the left divider, we have resistors R 1 and R 2; on the right divider, we have resistors R 3 and R4. The output voltage, V_out, is measured between the midpoints of these two dividers.
If the ratios of the resistors are chosen so that R 1 over R 2 equals R 4 over R 3, the two dividers produce exactly the same voltage at their midpoints, and V_out is zero. This is called the null mode — no signal at the output when the bridge is balanced.
Now, if one of the resistors — typically R 3 — is actually a sensor whose resistance changes slightly, say from R knot to R knot times (1 plus x), the bridge becomes unbalanced. This small change in R 3 produces a nonzero V out, even though the original background voltage was canceled out.
In deflection mode, we can write V out as the supply voltage V c c multiplied by the difference between the left divider ratio and the right divider ratio. Because the bridge starts balanced, even a very small change in R 3 creates a measurable output.
The advantage is that we've removed the large constant voltage and isolated only the change we care about. This makes it possible to apply high gain to that small signal without saturating the system.

slide26:
Up to now, we've looked at components that behave in a linear way — resistors, capacitors, and inductors — all of which follow the generalized Ohm's Law, voltage equals impedance times current.
But many real-world circuit elements are nonlinear components. Their voltage–current relationship does not form a straight line.
For example, a diode only allows significant current to flow in one direction once the forward voltage reaches a certain threshold — about 0.7 volts for silicon and 0.3 volts for germanium. In reverse bias, very little current flows until breakdown occurs.
Transistors, such as the NPN bipolar junction transistor shown here, can amplify

signals. By carefully biasing the base–emitter junction, a small change in base current controls a much larger change in collector current. This allows weak input signals to be amplified into stronger output signals.
We also have digital logic gates — OR, AND, XOR, NOR, NAND, and NOT — which are the building blocks of digital electronics. They process binary signals, producing specific outputs based on logical rules. By combining these gates, we can create flip-flops and sequential circuits, which store and manipulate digital information.
Nonlinear components make it possible to build amplifiers, oscillators, logic circuits, and microprocessors. They open the door to modern electronics — where linear analysis still plays a role, but must be combined with device-specific nonlinear behavior.

slide27:
Just as we can have linear systems and nonlinear systems, we also have nonlinear electronic components.
One of the most fundamental is the diode. A diode has a strong directional preference for current flow. In its forward direction, the current rises rapidly after a certain threshold voltage — about 0.7 volts for silicon. In the reverse direction, the current is extremely small until breakdown occurs. This is a clearly nonlinear voltage–current relationship.
If we combine two diodes in a specific configuration, we create a transistor. A transistor takes a small signal at its input and produces a much larger version of the same signal shape at its output — it acts as an amplifier. This ability to magnify weak signals makes the transistor an essential building block for electronics.
Multiple transistors can be combined to perform logic operations. For example, an AND gate outputs a high voltage only if both inputs are high. An OR gate outputs a high voltage if either input is high. More complex gates, like NAND, NOR, and XOR, can be built from these basic ideas.
By connecting logic gates, we can create circuits that add, multiply, store data, and make decisions — the very basis of digital computers.
This is why in electronics we need to understand both linear and nonlinear systems. Linear components handle predictable analog relationships, while nonlinear components give us amplification, switching, and the foundation for all digital processing.

slide28:
As we just discussed, we often want three things in a circuit:
very high input resistance,
very low output resistance,
and a large gain or magnification factor.
The operational amplifier — or op amp — delivers exactly that.
Inside an op amp are many transistors and other components, all arranged to meet these requirements. The internal circuitry can look complicated, but for analysis, we can use a simple model:
We apply a small input voltage, V in, between the non-inverting and inverting inputs. The op amp multiplies this by a large gain factor, G, to produce the output voltage, V out equals G times V in.
In the ideal model:
The gain is infinitely large.
The input impedance is infinite, so the op amp doesn’t draw current from the source — we capture the full signal.
The output impedance is zero, so the op amp can drive the next stage without loss.
Real op amps don’t reach these ideal values, but they can come close — for example, input impedance in the megaohm to teraohm range, and output impedance as low as a few hundred ohms.
An op amp is an active component — it needs its own power supply, typically labeled V S plus and V S minus. This internal power lets it amplify signals, something passive components like resistors, capacitors, and inductors cannot do on their own.

slide29:

Even though an operational amplifier is an active component, the basic circuit laws still apply — Kirchhoff’s Current Law and Kirchhoff’s Voltage Law always hold.
But for op amps, we have two special rules that make analysis much easier:
Rule one: The two input terminals are at the same voltage.#This happens because the op amp’s gain is extremely large. Even the tiniest difference between the inputs is amplified so much that the output adjusts to make the two inputs essentially equal in voltage.
Rule two: No current flows into either input terminal.#The input impedance of an op amp is extremely high — ideally infinite — so no measurable current enters the inputs.
These two rules are the foundation for almost all op amp circuit analysis. They let us treat the inputs as if they are at the same voltage and as if they draw no current, which simplifies solving even complex amplifier circuits.

slide30:
Let’s look at this inverting amplifier and walk through the analysis step-by-step using our two op amp rules.
First, the non-inverting input is connected to ground.#By rule one, the inverting input is also at zero volts — that’s our virtual ground.
By rule two, no current flows into the op amp input.#That means all the current coming through the input resistor R-i from the input voltage must flow through the feedback resistor R-f to the output.
Now, the current through R-i equals the input voltage minus zero, divided by R-i.#The current through R-f equals zero minus the output voltage, divided by R-f.
Since these currents are the same, we can say:#Input voltage divided by R-i equals negative output voltage divided by R-f.
If we multiply both sides by R-f, we get:#Output voltage equals negative R-f over R-i, times the input voltage.
The negative sign means the output is inverted compared to the input.#The size of the gain is the value of R-f divided by the value of R-i.
So, with just those two op amp rules, we can find the input–output relationship without worrying about the op amp’s internal complexity.

slide31:
Operational amplifiers can be connected in many different configurations, each producing a specific function.
Here you see several common modules:
A non-inverting amplifier, where the gain equals one plus the ratio of R2 over R1.
An inverting amplifier, where the gain equals negative R2 over R1.
A differentiator, which produces an output proportional to the rate of change of the input signal.
An integrator, which produces an output proportional to the time integral of the input signal.
A differential amplifier, which amplifies the difference between two input signals.
All of these can be analyzed using the same two op amp rules we discussed earlier: the inputs are at the same voltage, and no current flows into the input terminals.
While the details of each configuration can get more involved, for our purposes, it’s enough to understand the fundamental idea: by choosing how we connect resistors, capacitors, and feedback paths, we can make the op amp perform a wide range of analog signal processing tasks.
If you’re interested in the full derivations and formulas, the linked reference has complete explanations — but for this class, understanding the basic concepts is sufficient.

slide32:
We can also use operational amplifiers in measurement setups, particularly for impedance matching.
Looking into the input port of an op amp, the impedance is very high. This means it can capture a signal from a source without significantly loading it. On the other hand, when we take the signal from the output port, the impedance is very

low. This allows the signal to drive the next stage efficiently. In this way, the op amp satisfies the two opposing requirements we discussed earlier: high input impedance and low output impedance.
This principle applies to both DC and AC circuit analysis. Remember the voltage–current relationships for resistors, capacitors, and inductors. When you connect these components into a network, Kirchhoff's Voltage Law and Kirchhoff's Current Law give you enough equations to solve for all unknowns.
The beauty of phasor notation is that for AC steady-state circuits, you don't need to solve differential equations. You simply work with algebraic equations, just as in DC analysis. This is why, in a sense, we can say "AC is DC in disguise" — at least for steady-state analysis.
Of course, if you're dealing with transients — for example, when a signal is suddenly applied — you would still need to solve the time-domain differential equations. But once the system reaches steady state, algebraic methods are enough.
These circuit concepts are applied in many areas — from sensing signals in medical imaging to filtering unwanted noise. In fact, we've already seen examples where Fourier analysis helps us understand how circuits pass or block certain frequencies. This is crucial for designing data acquisition systems that can detect and process signals effectively.

slide33:
Let me take you back to an event from several years ago. We organized a symposium at the AAAS Annual Meeting — that's the American Association for the Advancement of Science. If you're familiar with Science magazine, that's actually the flagship publication of AAAS.
This meeting is one of the most important gatherings in the scientific community. It brings together experts from every field — medicine, engineering, mathematics, physics, education — you name it. The aim is not just to share research, but also to promote broad scientific understanding and collaboration.
Our symposium was titled X-ray Imaging Innovations for Biomedicine. We discussed advances in CT technology, covering both hardware improvements and algorithmic innovations. We also looked at how imaging can play a more predictive and personalized role in medicine.
The meeting is structured into many symposiums running in parallel. We had our session, shared our work, exchanged ideas, and learned from other fields. It was a valuable experience — not only for presenting our research, but also for gaining perspectives from outside our immediate domain.

slide34:
The day after our own symposium, I found myself free for the afternoon, so I decided to attend another session titled The Technology of Artificial Intelligence. It was led by Demis Hassabis, the co-founder and CEO of DeepMind — one of the world's leading AI companies, later acquired by Google.
I listened to several fascinating presentations. The speakers shared breakthroughs in AI research, from neuroscience-inspired architectures to game-playing systems that were pushing the limits of machine learning. The atmosphere was electric — you could feel that this was more than just incremental progress.
That afternoon convinced me of something important: AI had truly entered a revolutionary stage. The technology was no longer just an academic curiosity; it was becoming a powerful tool with transformative potential, ready to impact science, healthcare, and our understanding of the human mind. It was clear that an exciting new era was unfolding.

slide35:
That experience sparked an idea in my mind. Over the past few centuries, humanity has gone through several major transformations. We began with the Industrial Revolution — when machines took over physical labor and reshaped manufacturing, transportation, and the way we lived. Then came the Information Revolution, where computers and the internet allowed us to store, process, and share knowledge at unprecedented speed.
Now, with the rise of machine learning and artificial intelligence, we are entering what I like to call the Intelligence Revolution. This is not just about automating tasks — it's about creating machines that can learn, adapt, and make

decisions, performing tasks that were once thought to be exclusively human. This shift will influence every aspect of life — from science and medicine to education, communication, and beyond. It's a change as profound as the revolutions before it, but one that reaches directly into the realm of thought and reasoning.
When we look at the history of science, we can see that it has evolved through distinct paradigms over time. Thousands of years ago, science was primarily empirical. People described natural phenomena based on what they could directly observe — carefully recording patterns in nature without necessarily understanding the underlying mechanisms.
A few hundred years ago, the theoretical branch emerged. This was a major leap forward: scientists began building mathematical models and generalizations, allowing them not just to describe the world, but to predict and explain it.
In the past few decades, we've added a computational branch. With powerful computers, we can now simulate complex phenomena that are impossible to test directly, exploring theories in silico before ever stepping into a laboratory.
And today, we have entered a new era — the era of data exploration, often called eScience. Here, we unify theory, experiment, and simulation, but our driving force is data itself. Instruments and sensors capture enormous amounts of information, which is then processed by software and stored in vast databases. From there, we can apply data analytics, statistics, and increasingly, machine learning to extract patterns, features, and relationships directly from the data.
This approach — using big data and AI to uncover new knowledge — is so different that some call it the "fourth paradigm" of science. It represents a shift in thinking: instead of starting with a theory, we can sometimes let the data speak for itself.

slide36:
Here, we see the structure of a biological neuron — the basic building block of the nervous system.
A neuron has several main parts. The dendrites act like antennas, receiving incoming signals from other neurons. These signals travel toward the cell body, which contains the nucleus and all the essential cellular machinery to keep the neuron alive and functioning.
From the cell body, information is sent down a long projection called the axon. This is essentially the neuron's transmission cable, carrying electrical signals — known as action potentials — toward the axon terminals. Along the way, the signal may be sped up by segments of myelin sheath, which act as insulation.
At the axon terminals, the neuron communicates with its target cells — which could be other neurons, muscle fibers, or glands — across small gaps called synapses. Communication here happens chemically, through the release of neurotransmitters, or in some cases, electrically.
The important thing to remember is that neurons are not just passive wires. They integrate inputs, make decisions about whether to fire, and adapt over time. This biological model is what inspired the design of artificial neural networks in machine learning.

slide37:
Here is a simplified view of how neurons work. Think of this as the basic flow of information in our nervous system.
Incoming signals — whether from the outside world or from other neurons — arrive at the dendrites. These dendrites act as the input terminals of the neuron. They collect all incoming information, both excitatory and inhibitory.
If the combined strength of these signals is strong enough to cross a certain threshold, the neuron generates an electrical pulse called an action potential. This signal travels down the axon — the long fiber you see here — and moves toward the axon terminals.
At the axon terminals, the neuron communicates with the next cell, often through a synapse. There, chemical messengers or direct electrical connections pass the information forward, continuing the chain of processing in the nervous system.
This is the fundamental process behind everything from reflexes to complex thought — and it's also the biological inspiration for how artificial neural networks process information in AI systems.

slide38:
Now, here is the simplest way to think about it.
At the top, you see a biological neuron. It receives input through many branches, called dendrites. All of these signals are gathered in the main cell body, the soma. If the combined signal is strong enough, the neuron generates an electrical pulse that travels down the axon. Finally, it reaches the output interface — the synapse — where it passes the signal on to other neurons.
Below that is the mathematical model we use for an artificial neuron. Each input is represented as a variable, and not all inputs carry the same importance, so we assign a weight to each one. Larger weights mean that input has a bigger influence; smaller weights mean less influence.
The neuron first performs a linear operation — a weighted sum of all the inputs, which is essentially an inner product. But that's not enough. Just like in biology, small random fluctuations — small "nudges" — are ignored. Only when the total input crosses a certain threshold does the neuron respond strongly. This is where the nonlinear transformation comes in, producing the actual output.
So, in short: first we sum the inputs in a weighted way, then we pass the result through a nonlinear function that decides whether to respond — and how strongly.

slide39:
Now let's look at how we represent a neuron in a computer — what we call a perceptron.
A perceptron takes multiple inputs, each labeled here as a 1, a 2, a 3,…, a n. Each input has a weight, shown as w 1,w 2,w 3,…,w n, which controls how important that input is.
The first thing we do is multiply each input by its weight. Then we sum them all together — this is an inner product operation. Often, we also add an extra term called a bias, b, which shifts the output. You can think of the bias as an offset — a way to control the baseline response of the neuron, even if all the inputs are zero.
Once we have this weighted sum, we pass it through a nonlinear function, which we usually call the activation function. Here it's labeled sigma. This step is essential — without it, the perceptron would just be a simple linear device.
There are many choices for the activation function. A common one is the sigmoid, which produces an S-shaped curve. Another is the hyperbolic tangent, or tanh, which gives outputs between minus one and plus one. There are also piecewise-linear functions such as the ReLU — the rectified linear unit — where the output is zero for negative inputs and increases proportionally for positive inputs.
So in short: a perceptron takes weighted inputs, adds them up, applies a bias, and then uses a nonlinear transformation to produce its output. This artificial neuron is the basic building block for artificial neural networks — just as resistors, capacitors, and inductors are building blocks in electrical circuits.

slide40:
Now that we understand a single perceptron, let's see what happens when we connect many of them. This gives us an artificial neural network.
At the far left, we have the input layer — each circle here receives one piece of information from our data. These could be pixel values from an image, measurements from a sensor, or any other kind of features we want the network to process.
Next, we have one or more hidden layers. Each neuron in a hidden layer takes the outputs from the previous layer, applies its weights, sums them up, passes the result through its activation function, and then sends the output to the next layer. These hidden layers are where the network learns increasingly complex patterns.
Finally, at the far right, we have the output layer. In a classification task, each neuron here might represent a possible label — for example, "dog," "cat," or "car." The neuron with the highest output value would be the network's prediction.
Here's the key idea:
A single neuron can only separate very simple patterns.
By stacking many layers, the network can build up a hierarchy of features.

The first layer might detect basic edges or colors.
The next layer might detect shapes like wheels or ears.
Higher layers combine these into entire objects, like a car or a cat.
Training the network works like this: we start with random weights, so the predictions are essentially guesses. We compare the network's output to the correct answer and compute the error. Then, using a process called backpropagation, we send this error backward through the network, adjusting the weights slightly to reduce the error next time.
With enough data, enough layers, and many training cycles, the network learns to recognize very complex patterns — the kind of capability that powers things like self-driving cars, medical image analysis, and voice recognition.

slide41:
Once you have the basic idea of how a neural network works, you can imagine that there's no single way to connect the neurons. In fact, researchers have developed many different network topologies, each suited for specific tasks.
Here in this chart, you can see a variety of examples.
At the simplest level, we have the perceptron and feed-forward networks, where information flows in one direction from input to output.
Then there are radial basis networks and more specialized architectures like recurrent neural networks or RNNs, which loop information back so the network can remember past inputs.
Long Short-Term Memory networks — LSTMs — and Gated Recurrent Units, or GRUs, are powerful variants for processing sequences, such as speech or time-series data.
Other designs, like autoencoders and variational autoencoders, are used for compressing data and then reconstructing it, often to find hidden patterns or generate new examples.
You also see Boltzmann machines, Hopfield networks, and deep belief networks, which are useful for certain types of learning and pattern recognition.
The key point is that these different architectures are like tools in a toolbox. Some are better for images, some for language, some for prediction over time.
Once you understand the fundamentals — inputs, weights, activation functions, and training — all these variations are just different ways of wiring those same building blocks together.

slide42:
It's important to realize that not every neural network configuration will be equally useful. Designing an effective network requires engineering insight — the architecture has to match the nature of the problem.
For example, a simple feed-forward structure can be thought of as a kind of multi-scale analysis, similar to how wavelet transforms work in signal processing. If your data has features at different scales, you need a structure that can capture them.
Convolutional operations, widely used in image and signal processing, are rooted in the mathematics of linear systems. They're excellent for recognizing local patterns that repeat across space or time.
Shortcut connections and feedback loops, inspired by control theory, can help stabilize training and improve learning in very deep networks.
In some cases, we even borrow ideas from game theory — for example, adversarial mechanisms like GANs, where two networks compete to improve each other.
And when we're working with oscillations, waveforms, or electromagnetic signals, concepts from complex analysis — such as amplitude and phase — can be built directly into the network using complex-valued weights.
The takeaway here is that neural network design is not guesswork. It's about choosing the right principles from mathematics and engineering, and embedding them into the architecture so that it's well-matched to the problem at hand.

slide43:
Now, how do we actually optimize a neural network so that it performs well?#As I mentioned earlier, when you start with a fresh network, its performance is usually terrible — the outputs are essentially random. The idea is to adjust the parameters — the weights, which we represent as a vector w — in such a way that the network's predictions get closer and closer to the desired results.

We measure the difference between the network’s current output and the target output using what we call a loss function, or an error function. The goal is to make this error as small as possible.
One common approach is gradient descent. We start with an initial set of weights — often chosen randomly. Then, in each training step, we look at the slope of the error function with respect to each weight. This slope, or gradient, tells us which direction increases the error — so we move in the opposite direction, reducing the error.
Mathematically, the new weight vector equals the old weight vector, minus a small fraction of the gradient. That fraction is controlled by the learning rate, a simple scaling factor between zero and one. If the learning rate is too big, the updates may overshoot, and the training becomes unstable. If it’s too small, the network learns very slowly.
By repeatedly updating the weights in this way, the error decreases step by step. Eventually, if everything is set up well, the process converges and the network produces accurate outputs.

slide44:
Once you understand the concepts, you might want to try them out in practice. MATLAB now offers a Machine Learning Toolbox — a powerful package that helps you perform parameter optimization, data clustering, and pattern recognition.
With this toolbox, you can access a variety of ready-made functions for tasks like feature extraction, classification, and regression. It also comes with tutorials, application notes, and built-in demos, so you can learn by experimenting.
For example, you can try leaf recognition, fingerprint detection, or even music genre classification — all with the same framework. The advantage is that you don’t have to start coding everything from scratch. Instead, you can focus on understanding the algorithms, adjusting parameters, and seeing how those changes affect performance.
This makes it an excellent platform for quickly testing ideas, prototyping solutions, and getting a feel for how machine learning techniques behave with real data.

slide45:
Once we define a loss function — the measure of difference between the network’s output and the ground truth — our goal is to reduce that difference step by step. We do this by adjusting the network’s parameters in a way that moves us downhill on the loss surface.
You can imagine this process like standing on a hilly landscape and trying to reach the lowest valley. At each step, we look around, find the steepest downward direction, and take a small move in that direction. This is the essence of gradient descent.
Over time, with each update, the loss gets smaller and smaller until the network converges to a good solution — ideally, a point where further changes no longer improve the performance.

slide46:
Here, you see a visual demonstration from an online animation that shows how a neural network learns during training. Backpropagation is the key idea behind this process.
Think of it like this — the network makes a guess, compares that guess to the correct answer, and measures the error. Then, instead of adjusting everything randomly, it calculates how much each parameter contributed to that error. Using this information, it sends a correction signal backward through the layers, adjusting the parameters step by step.
With enough training cycles, the network gradually reshapes its decision boundaries, as you see in the animation, until it can classify almost all points correctly.
If you’re curious, I encourage you to watch the video linked here to see backpropagation in action.

slide47:
This is an article I wrote for IEEE Access titled A Perspective on Deep Imaging.

As imaging scientists, we've traditionally thought of images as something you start with to process or analyze. But in medical imaging, it's the opposite — the image is the result of a long reconstruction process.
What's exciting is that with machine learning, we can rethink that process entirely. Instead of following the conventional pipeline step by step, we can train neural networks to directly generate high-quality images from the raw data itself. This opens the door to faster reconstructions, better image quality, and even reduced radiation dose in certain applications.
In short, deep learning isn't just another tool in the box — it has the potential to transform the way we produce and interpret medical images.
Now let's look at one practical example — improving the signal-to-noise ratio in images.
In medical imaging, noise is inevitable. It can come from the physics of the scanner, from the patient's movement, or from trying to keep the radiation dose low. Traditionally, we use filtering techniques to clean up an image, but these often blur fine details along with the noise.
With modern machine learning, we can do better. By training neural networks on large datasets — for example, the thousands of scans acquired every day in hospitals — the system can learn what a "clean" image should look like, without erasing important diagnostic details. Instead of throwing those routine scans away, we can use them to teach the network.
As a result, we can transform a noisy scan into one that is much clearer, improving both diagnostic confidence and patient safety.
So that's the big picture — and we'll wrap up here for today.

slide48:
Before we close, let me show you a roadmap that pulls together the foundational ideas we've been discussing.
This diagram is a kind of "big picture" for the mathematical and engineering concepts underlying modern imaging and machine learning. It includes complex numbers, Fourier transforms, convolution, and sampling theory — all of which form the language we use to describe signals and images.
You can see how these elements connect: from basic mathematics like Euler's formula, to signal operations such as convolution and correlation, to the Fourier series and transforms that let us work in the frequency domain. We also have system models from electrical engineering, like impedance and circuit laws, and finally, neural network building blocks — showing how the same principles extend into modern AI.
Think of this as your conceptual map. Each box here represents a tool in your toolbox, and together, they form the foundation for advanced methods in imaging, machine learning, and beyond.

slide49:
So here, you can see that all of these relationships are brought together in one poster.
Some of the diagrams—like the one we have here—actually exist in several different published versions. In this case, I've redrawn the figure myself to avoid any copyright issues.
For a classroom lecture, using the original version is generally fine. But if you eventually want to publish your work, you either need to get formal permission to use those copyrighted figures or you should redraw them in your own style so they're fully original.

slide50:
For your homework, I'd like you to work through this sensitivity analysis problem.
The idea is to determine how to choose the resistor values so that the circuit's sensitivity is as high as possible. Sensitivity here is defined as the derivative of V out with respect to x, where x represents a small fractional change in the sensor resistance.
In other words, if x changes just a little, we want V out to change as much as possible. That's what we mean by "high sensitivity."
You'll be able to adjust both R L and R S. If you work through the math, you'll find that the sensitivity reaches its maximum when the two resistances are

equal.