Explanatory Notes for 6.390

Shaunticlair Ruiz (Current TA)

Spring 2023

Many Layers

We are finally ready to build our **complete** neural network. We'll just retrace the steps of the 2-layer case.

Notation 1

The total **number** of **layers** in our **neural network** is notated as L.

Typically we notate an **arbitrary** layer as ℓ (or l).

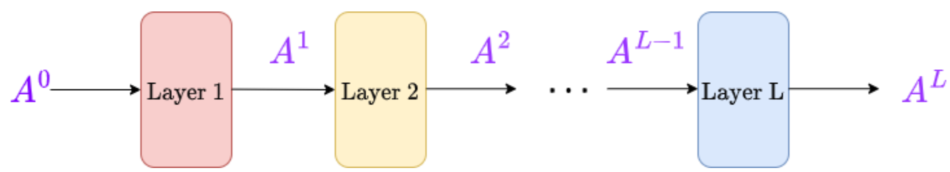
Since x is, for all purposes, **equivalent** to a vector A, we will call it A^0.

Notation 2

Our **neural network**'s input x is used in the **same** way as every term A^ℓ.

So, we will **represent** it as

$$x = A^0$$



Again, we see that the **output** of layer ℓ is the **input** of layer $\ell + 1$.

Concept 3

Each layer **feeds** into the next layer.

A^ℓ is the **output** of layer ℓ, and the **input** of layer $\ell + 1$.

This means that the **output** dimension must **match** the next **input** dimension.

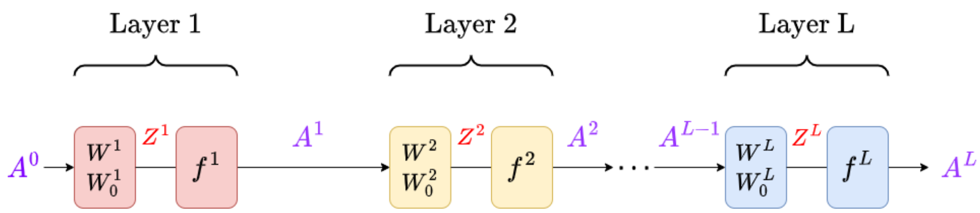
$$\overbrace{n^\ell}^{\text{Output}} = \overbrace{m^{\ell+1}}^{\text{Output}}$$

And the **dimension** of A^ℓ is $(n^\ell \times 1) = (m^{\ell+1} \times 1)$.

Our Complete Neural Network

We can break our layers into components, so we can see the functions involved.

With this, we build our final neural network:



With this, we can see how each layer is **related** to each other: as we **mentioned**, the **output** of one layer is the **input** of the next layer.

Here is the computation we do for layer ℓ:

Key Equation 4

The calculations done by layer ℓ are given by

$$Z^{\ell} = (W^{\ell})^{T} A^{\ell-1} + W_0^{\ell}$$

and

$$A^{\ell} = f(Z^{\ell})$$

Which combine into:

$$A^{\ell} = f(Z^{\ell}) = f\left((W^{\ell})^{T} A^{\ell-1} + W_0^{\ell} \right)$$

Hidden Layers and the "First Layer"

Now that we have a full network, we introduce some useful vocab.

Definition 5

A **hidden layer** is any functional layer except for the **output** (last) layer.

It is called a "**hidden**" layer because, if you're viewing the whole neural network based on

- **Input** x (first input)
- **Output** A^L (final output)

You can't see **hidden layers** from outside the network.

Based on this definition,the **number of hidden layers** in a network is the layer count, minus one: $L - 1$.

Note that there's one point of confusion: online, you may see that the hidden layer is "any layer other than the **input** (first) or **output** (last) layer".

This is because, often, we consider the input itself to be a separate "**input layer**".

Despite this fact, when someone counts the number of layers in a neural network, they're usually only counting the hidden and output layers: we **don't count** the input layer.

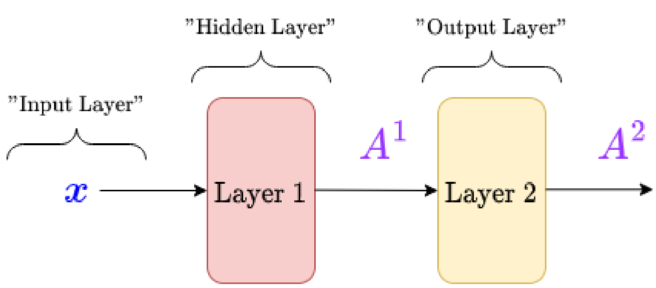
It confused me, too.

Definition 6

The **input layer** is a layer that brings the **input** into the network. It applies **no functions** to the data.

Because the input layer has **no effect** on our data (it just moves it), we **don't count the input layer** when we're saying how **many layers** a network has.

Example: Consider the following network from earlier:



In this network, x is passed into the network by the **input layer**. This layer is **before** layer 1 (you could think of it as "Layer 0").

Despite having the input layer, plus layer 1 and 2, we count only

- **Two** layers in our network:

– One hidden layer: Layer 1.

– One output layer: Layer 2.