# Explanatory Notes for 6.390

Shaunticlair Ruiz (Current TA)

Fall 2022

## The weakness of a single layer

What can we do with a single layer? Well, our **LLC** model gives us an example: it has the **nonlinear** sigmoid activation, but acts as a **linear** separator.

Why is that? Why is the separator still linear, if the **activation** isn't?

Well, let's take the **linear** separator created by the pre-activation:

$$z = w^T x + w_0 = 0 \tag{1}$$

This is our **boundary** for just a linear function. But adding the nonlinear activation should make it more **complex**, right?

Well, it turns out, we can represent our **activation** boundary with a **linear** boundary.

**Example:** Continue our LLC example. If $z = 0$, then $\sigma(z) = \sigma(0)$. Our boundary is

$$\sigma(z) = \sigma(0) = \frac{1}{2} \tag{2}$$

Wait. But that means that $\sigma(z) = .5$ is the same as $z = 0$: the same inputs $x$ cause both of them, so they have the same boundary!

$$\text{Linear boundary } z = 0 \iff f(z) = \frac{1}{2} \tag{3}$$

Summary:

- $\sigma(z) = .5$ is the **same** as $z = 0$.
- $z = 0$ is **linear**.
- Thus, our sigmoid boundary is **linear**.

We can apply this to other activation functions. In general, any constant boundary for most $f(z)$ is equivalent to some linear boundary $z = C$:

Assuming that $f$ is invertible, which it often is.

$$z = C \qquad \iff \qquad f(z) = f(C) \tag{4}$$

Since $z = C$ is linear, we know that our activation separator $f(x) = f(C)$ is linear too.

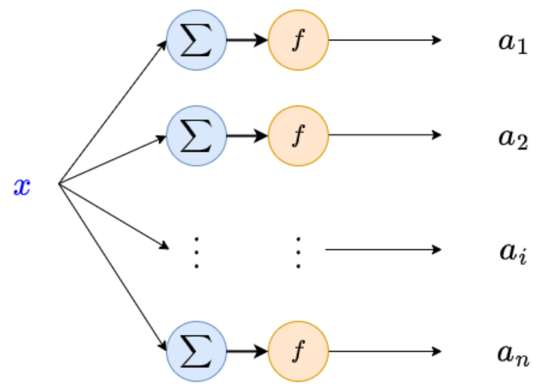
**Concept 1**

A single neuron creates a **linear separator**, even if it has a **nonlinear** activation.

This is because any **boundary** for $f(z)$ we can create, can be represented by some **linear** boundary in $z$.

It turns out, adding more neurons **within** the layer doesn't change much: because they act in **parallel**, each neuron acts separately, and the things we said above are still **true** for each output $a_i$.

There are exceptions, but this is true for most useful activation functions.




Each of these neurons has the same input, $x$.

So, in order to create nonlinear behavior, we need at least two layers of neurons in **series**.

So, we'll start **stacking** layers on each other: each layer **feeds** into the next one.

**Concept 2**

A **single layer** of neurons has **linear** behavior.

We need **multiple** layers to get a nonlinear **neural network**.