# Explanatory Notes for 6.390

Shaunticlair Ruiz (Current TA)

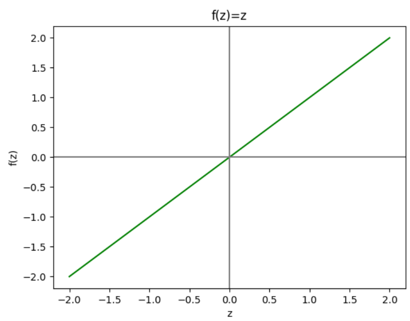
Spring 2023

## Example of Activation Functions

So, let's look at some possible **activation** functions:

- **Identity** function $z$:

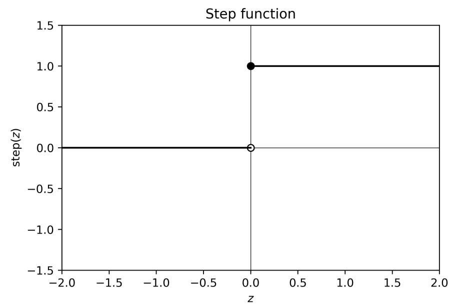
$$f(z) = z \tag{1}$$




  - This function is called an **identity** function because it "preserves the identity" of the input: the output is the same.
  - This is an example of a **linear** function.
    * As we described in the last section*, linear activation can't make our model more **expressive**.
    * So, we **almost never** use it (or any other **linear** function) as an activation for a **hidden** layer.
  - We mainly use this an **output** activation function: it allows our final output to be any real number.
    * This is a good activation function for a **regression** model, which returns a **real** number.
    * It's a simple function, that can return **any** real number. By constrast, sigmoid and ReLU both have **limited** output ranges.

- **Step** function $\text{step}(z)$:

$$\text{step}(z) = \begin{cases} 1 & \text{if } z \geqslant 0 \\ 0 & \text{if } z < 0 \end{cases} \tag{2}$$



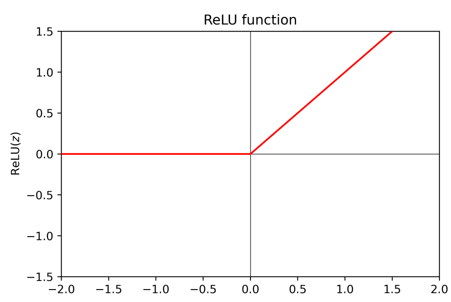

- This function is basically a **sign** function, but uses $\{0, 1\}$ instead of $\{-1, +1\}$.

- Step functions were a common early choice, but because they have a **zero** gradient, we can't use **gradient descent**, and so we basically **never** use them.

Same reason we replaced the sign function with sigmoid.

- **Rectified Linear Unit** $\text{ReLU}(z)$:

$$\text{ReLU}(z) = \max(0, z) = \begin{cases} z & \text{if } z \geqslant 0 \\ 0 & \text{if } z < 0 \end{cases} \tag{3}$$



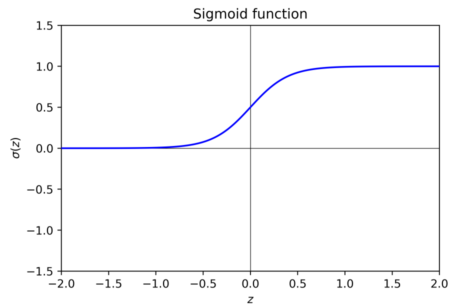

- This is a very **common** choice for activation function, even though the derivative is undefined at 0.

- We specifically use it for internal ("**hidden**") layers: layers that are neither the **first** nor **last** layer.

They're "hidden" because they aren't visible to the input or output.

- **Sigmoid** function $\sigma(z)$:

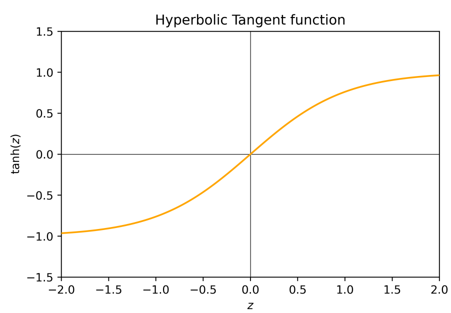
$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{4}$$




- This is the **activation** function for our **LLC** neuron from before.
- Just like LLC, it's useful for the **output neuron** in **binary classification**.
- Can be interpreted as the **probability** of a positive $(+1)$ binary classification.
- We can also use this for multiclass when classes are **NOT** disjoint: we use one sigmoid per class.
  * Each sigmoid tells us how likely the data point is to be in that class.

• **Hyperbolic Tangent** $\tanh(z)$:

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{5}$$




- This is function looks similar to sigmoid over a different **range**.
- Unfortunately, it will not get much use in this class.

• **Softmax** function $\text{softmax}(z)$:

$$\text{softmax}(z) = \begin{bmatrix} \exp(z_1)/\sum_i \exp(z_i) \\ \vdots \\ \exp(z_n)/\sum_i \exp(z_i) \end{bmatrix} \tag{6}$$

- Behaves a like a **multi-class** version of **sigmoid**.
- Appropriately, we use it as the **output neuron** for **multi-class** classification.

  - Can be interpreted as the **probability** of our $k$ possible classifications.
    * "Disjoint" probability: each option is separate. Sum of the rows adds up to 1.

**Concept 1**

For the different **activation functions**:

- $f(z) = z$ isn't used for **hidden** layers, but we can use it for regression **output**.
- $\text{sign}(z)$ is **rarely** used.
- $\text{ReLU}(z)$ is often used for "**hidden**" layers.
- $\sigma(z)$ is often used as the **output** for **binary classification**.
- $\text{softmax}(z)$ is often used as the **output** for **multi-class classification**

$\tanh(z)$ is useful, but not a focus of this class.

Remember this caveat, though:

**Clarification 2**

Multi-class depends on whether a **data point** can be in **multiple classes at the same time**.

- $\text{softmax}(z)$ assumes our classes are **disjoint**: you can only be in **one** class.
  - This is usually what people mean by **multi-class**.
- $\sigma(z)$ can be used when classes are **not disjoint**: you can be in **multiple** classes.
  - You can think of this as **binary classification** for each class.

When using sigmoids, we need **one** sigmoid for each **class**.

**Example:** We can compare use cases for each of these:

- Softmax could be used to answer, "which word is the next one in the sentence?"
  - Every word in a sentence is only followed by one word: they're mutually exclusive.
- Sigmoids could be used to answer, "what genre of book is that?"
  - A book is often in more than one genre.

## Loss functions and activation functions

As we can see above, your **activation** function depends on what kind of **problem** you're dealing with.

The same is true for our **loss** function: we used **different** loss functions for classification and regression.

Classification can be further broken up into **binary** versus **multiclass** classification.

To summarize our findings, we'll **sort** this information:

**Concept 3**

Each of our **tasks** requires a different **loss** and output **activation** function.

We emphasize that we specifically mean the output activation function: the activation function used in **hidden layers** doesn't have to match the loss function.

| task | $f^L$ | | Loss | |
|---|---|---|---|---|
| Regression | Linear | $z$ | Squared | $(g-y)^2$ |
| Binary Class | Sigmoid | $\sigma(z)$ | NLL | $y \log g + (1-y) \log(1-g)$ |
| Multi-Class | Softmax | $\text{softmax}(z)$ | NLLM | $\sum_j y_j \log(g_j)$ |

**Special Case**: If we allow **multiple** classes at the **same** time (non-disjoint), we use **binary** classification for each of them, rather than multi-class.

**Example:** An example for each type:

- **Regression**: Predicting the amount of rainfall in centimeters tomorrow.
- **Binary Classification**: Will the stock market go up or down tomorrow?
- **Multi-Class**: What species of tree is this?
- **Multiple Binary**: What are the themes in this movie?

## Other Considerations

You might consider using other functions, based on the needs of a more specialized task. We'll ignore those cases, for the most part.

But, if you want to try a new function, the **data type** is the most important for whether we can use it.

**Concept 4**

If you want to use a new **activation** or **loss** function, you have to pay attention to the **input/output** type.

**Example:** $\tanh(z)$ outputs over the range $(-1, 1)$. We could use it, if that was the range we wanted.

Be careful, though:

**Clarification 5**

It's important to stress that while our **output activation** depends on the task, **hidden layers** don't have to.

Hidden layers can use one of several **different** activation functions, regardless of the **task**.

However, some activation functions tend to be **better** for making a model than others.

**Example:** Often, we use ReLU for hidden layers, but it's rarely used as an output activation function.

We also might use **sigmoid** as a hidden layer for a regression model, even though regression most commonly uses a **linear** output.