

**CME 216, ME 343 - Spring 2020**

**Eric Darve, ICME**



As we have seen previously, supervised machine learning is often concerned with two different tasks:

- classification: the output is a category or integer value
- regression: the output is a real number or a real-valued vector.

For DNN, there are common architectures that are used for each case.

Regression will be mostly our focus.

Generally the last layer does not have a non-linear function applied to it.

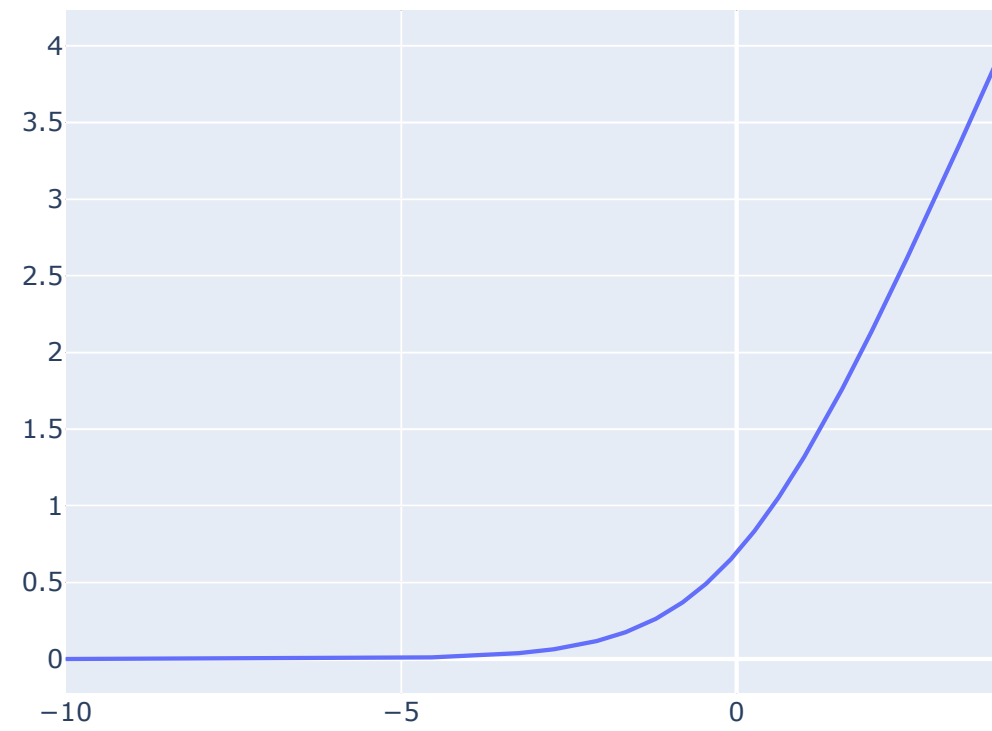
That is  $\phi(x) = x$ .

Depending on the situation, various "tricks" can be used to enforce certain conditions.

For example if you want the output to be positive it is common to apply a `softplus` activation for the last layer:

$$\text{softplus}(z) = \log(1 + \exp(z))$$

softplus function





$$\text{softplus}(z) = \log(1 + \exp(z))$$

You can see that when  $z$  is negative, softplus is close to 0.

When  $z$  is positive and large, softplus is close to  $z$ .

The sigmoid and tanh functions can be used if you know that the output is bounded.

In that case, you can shift and rescale your data so that it falls in the  $[0,1]$  interval (sigmoid), or  $[-1,1]$  interval for tanh.

The typical loss function for regression is the mean squared error.

Another option is the mean absolute error, but this loss function is more difficult to train with since it is not differentiable at 0.

The mean squared error tends to emphasize training points where the error is large.

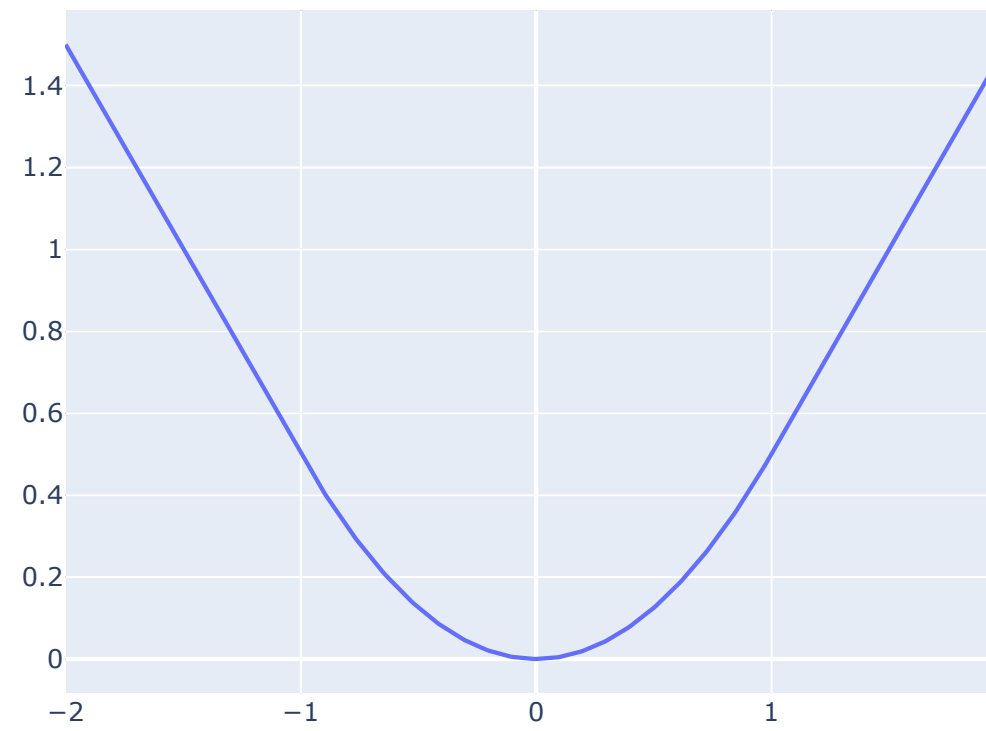
Mean absolute error gives about weight to all training points.

A compromise is the Huber loss defined as

$$x^2/2, \text{ if } |x| \leq \delta$$

$$\delta(|x| - \delta/2) \text{ otherwise}$$

Huber function



The second task is classification.

Let us assume that for example we are trying to classify images and label them with the object depicted.

Categories may be: car, horse, boat, truck, person.



We have 5 categories that are going to be represented by an integer between 0 and 4.

The way to represent this is to apply the softmax function to the output.

Given a vector of output  $z_i$ , the softmax function is

$$\frac{\exp(z_i)}{\sum_{j=1}^{n_{\text{categories}}} \exp(z_j)}$$

$n_{\text{categories}}$  is the number of categories or labels.

The vector softmax sums up to 1 and all its entries are positive.

So this can be interpreted as a probability vector.

$$a_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

$$\sum_i a_i = 1 \text{ and } a_i \geq 0$$

$$a_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

The index  $i$  for which  $a_i$  is the largest is interpreted as the category guessed by the DNN.

In the tensorflow jargon,  $z_i$  are called logits while  $a_i$  are the probabilities for the categories.

To calculate the loss function, we need to define cross-entropy, which can be used to measure the distance between two probability distributions.