1 W1

1.0.1 Lecture

$ReLu(x) = max(0, x)$ - function used sometimes for perceptrones.

We first need to decide on the dependencies between data and output. Example with house price:

- School quality
- Number of rooms
- Zip code (wat?)

From these we can derive:

- Wealth
- Walkability

By stacking together a few of the single neurons of the simple predictors we can obtain a more complex network that will actually do something. All you need to give is the input x and the output y for a number of examples in your training set and all those things in the middle will figure out by themselves(no).

Each layer recognizes features or bypasses them from the previous layer.

More examples for supervised:

- Image recognition(CNN convolution)
- AD prediction(CNN)
- translation(RNN))

- audio recognition(RNN recurrent)

There are two types of data: structured(tables) and unstructured(audio, images).

What happend to our society over the last ten years is that we've collected a huge amount of data due to the strong digitization. There are a lot of stuff in the digital realm.

Turned out that neural networks performance becomes better as amount of data grows comparing to any other method.

Switching from sigmoid to ReLu made gradient descent converge much faster.

Faster iterations -> more discoveries.

1.0.2 Geoffrey Hinton interview

He heard that brain stores memories as holograms in 1966(WAT? is there anything in cs that wasn't discovered in 50s 60s?)

Took physics and physiology, then philosophy and psychology.

Eventually got a PhD in AI. Couldn't find a job in Britain. Went to California(there Don Norman and David Rumelhart). In early 1982 He, David Rumelhart and Ron Williams developed the backprop algorithm, although this was discovered earlier by David Parker. Paul Werbos had published it even earlier but didn't get attention. In their paper they developed an algorithm for training network on family tree, so it can predict the third word in sentences like "Mary has mother".

He finds his work with Terry Sejnowski on Bolzman machines the most interesting. The more practical method is restricted Bolzman machines.

note: generative adversarial nets

2 W2

2.0.1 Lecture

given $x = \{x_1..x_n\}, w = \{w_1..w_n\}$ and $b \subset \Re$

Linear regression: $y =< w^T, x > +b$

Logistic regression: $y = P(y = 1|x)$

$y = P(y = 1|x) = \sigma(< w^T, x > +b)$ - a hyperplane separates two subspaces and then applies non linear transform. Sigmoid function has a region that goes from zero and grows almost linearly then becomes almost one. The points in this region determine the learning process, the points that are classified as 0 or 1 do not affect learning because the derivative is almost zero as function approaches 1 or 0. This problem resulted in the introduction of ReLu function.

So multiple layer network consists of layers that separate linearly some features in the data obtained in the previous layer. Also interesting point is that two layer network is capable of classifying convex areas(polytops) formed by hyperplanes and three layer network can classify the intersection of those convex shapes.

The loss function. $L = \frac{1}{2}(y - y_{true})^2$ is good but leads to non convex surfaces. $L = -(y_{true} * \ln(y) - (1 - y_{true}) * \ln(1 - y))$ is better(no, well sometimes). Also it is useful to add wights to the loss function to regularize.

Something about derivatives and convexity.

- `https://en.wikipedia.org/wiki/Convex_function`
- `https://en.wikipedia.org/wiki/Convex_optimization`
- `https://en.wikipedia.org/wiki/Convex_polytope`