Data, Environment and Society: Lecture 25: Neural Networks

Instructor: Duncan Callaway
GSI: Salma Elmallah

November 26, 2019

Announcements

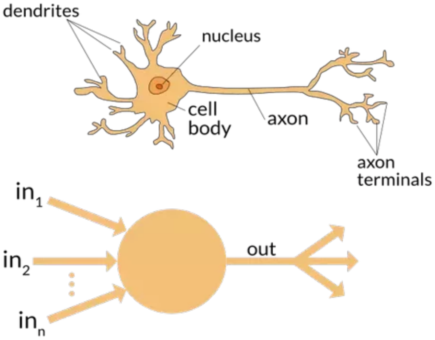
-

Today's outline

- Neural networks (NN) – very brief introduction
- Experiment with tensorflow playground – try fitting different classification problems
- Exam handout and discussion

Neural networks: Origins

- The name is due to analogy with brains
  - But the original purpose was not to reproduce cognition
- First developed in 1943
- Little research activity ~1960-1990's due to computing limitations
  - Major exception: Werbos developed back-propogation in 1974. First effort to get NN to "learn" parameters
- Computing advances made "deep" NN possible in the last 20 years



Mathematics for a single “neuron”

In words, each neuron...

- Takes a vector of values as inputs
- Creates a scalar from a linear combination of the vector entries
- Passes the resulting scalar through an “activation function”
- Outputs a single value from that activation function

Terminology analogies:

- Electrical signal from other cells $\rightleftarrows$ input
- Neuron $\rightleftarrows$ Activation function
- Electrical signal to other cells $\rightleftarrows$ output

Mathematics for a single "neuron"

In words, each neuron...

- Takes a vector of values as inputs
- Creates a scalar from a linear combination of the vector entries
- Passes the resulting scalar through an "activation function"
- Outputs a single value from that activation function

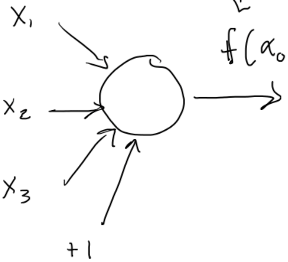
Terminology analogies:

- Electrical signal from other cells $\rightleftarrows$ input
- Neuron $\rightleftarrows$ Activation function
- Electrical signal to other cells $\rightleftarrows$ output

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

activation f'n

$$f(\alpha_0 + \alpha^T x)$$

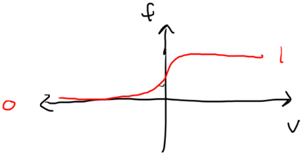


What's f, the activation function?

1. sigmoid
2. $\tanh$
3. rectified linear

What's f, the activation function?

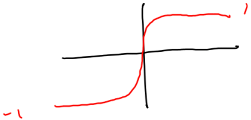
1. sigmoid



$v = \alpha_0 + \alpha^T x$

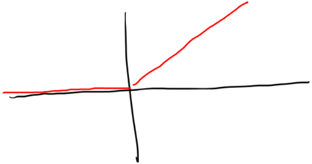
$$f(v) = \frac{1}{1+e^{-v}}$$

2. tanh



$$f(v) = \frac{e^{v} - e^{-v}}{e^{v} + e^{-v}}$$

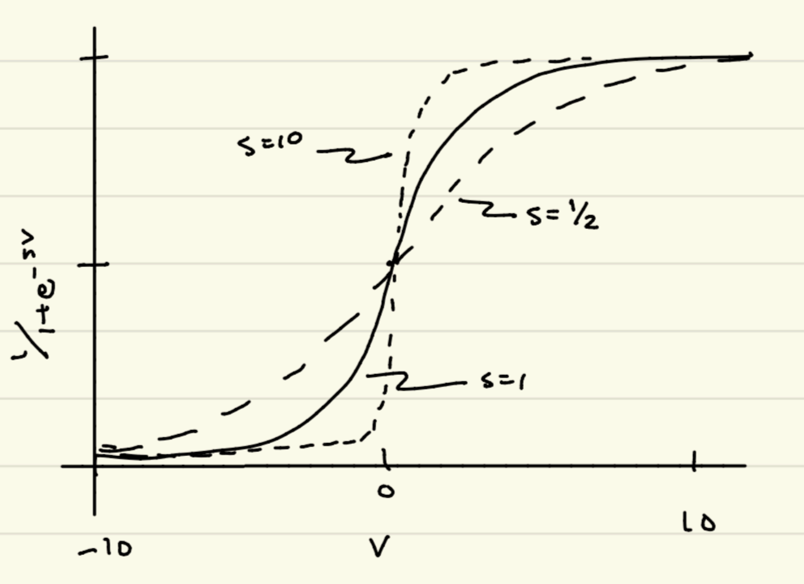
3. rectified linear



$\max(0, v)$

How the sigmoid function works

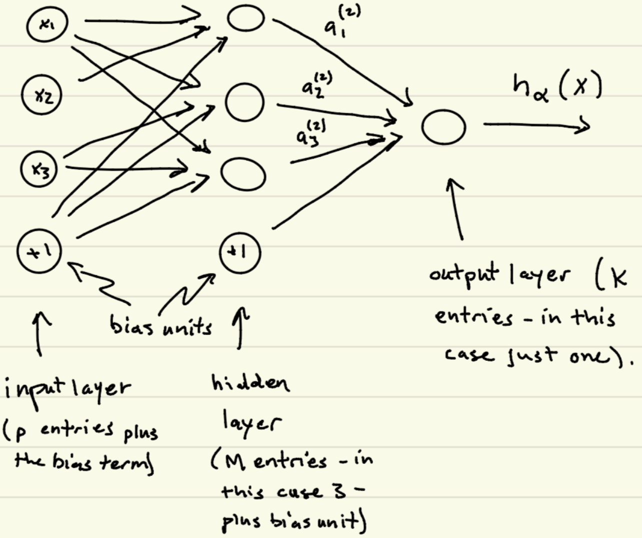
How the sigmoid function works



Remember: $v = \alpha_0 + \alpha^T \mathbf{x}$

Neural network: just gang the neurons together

Neural network: just gang the neurons together



Convention: $a_i^{(\ell)}$ is the output of the i^{th} neuron in the ℓ^{th} layer.

Mathematical merging of neurons

Convention:

- $\alpha_{ij}^{(l)} \rightarrow$ weight from node j in layer l to node i in $l+1$ layer.
- $a_i^{(l)} \rightarrow$ output of node i in layer l.
- $z_i^{(l)} \rightarrow$ input into node i in layer l.

$$a_1^{(2)} =$$

$$a_1^{(3)} =$$

Question: What are the parameters of a neural network model?

Mathematical merging of neurons

Convention:

- $\alpha_{ij}^{(l)} \to$ weight from node j in layer l to node i in $l+1$ layer.
- $a_i^{(l)} \to$ output of node i in layer l.
- $z_i^{(l)} \to$ input into node i in layer l.

$$a_1^{(2)} = \alpha_{10}^{(1)} + \alpha_{11}^{(1)} a_1^{(1)} + \alpha_{12}^{(1)} a_2^{(1)} + \alpha_{13}^{(1)} a_3^{(1)}$$

$$a_1^{(3)} = \alpha_{10}^{(2)} + \sum_{j=1}^{M} \alpha_{1j}^{(2)}$$

Question: What are the parameters of a neural network model?
Just the α values. a values are outputs from internal nodes or neurons. We call these "hidden states" because they depend on the input values x.

Thinking about the features and target

Let's watch this video. It uses graphics in a nice way to explain what NNs are doing.

`https://www.youtube.com/watch?v=aircAruvnKk`

Start the video at 2:05. We'll stop watching around 5:30.

Compact notation motivates a name...

Compact notation motivates a name...

$$
\begin{aligned}
x &: \text{vector of inputs} \\
z^{(2)} &= \alpha_0^{(1)} + \alpha^{(1)} x \\
a^{(2)} &= f(z^{(2)}) \\
z^{(3)} &= \alpha_0^{(2)} + \alpha^{(2)} a^{(2)} \\
&\vdots \\
h_\alpha(x) &= f(z^{(\ell)}) \quad \ell \text{ is the number of layers}
\end{aligned}
$$

For this reason we call these networks "feedforward" networks because information passes from the features (predictors) to the output (target)

Fitting the model

Training data: $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$

$x_k \in \mathbb{R}^p$ (p features)
$y_k \in \mathbb{R}^K$ (k outputs)
Objective function:

Fitting the model

Training data: $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$

$x_k \in \mathbb{R}^p$ (p features)
$y_k \in \mathbb{R}^K$ (k outputs)
Objective function:

$$J(\alpha; x_k, y_k) = \frac{1}{2} \| h_\alpha(x_k) - y_k \|^2$$

output of NN.

$\| \cdot \|^2$ is the "two norm" i.e. $\sum_{i=1}^{n} a_i^2$

n_ℓ = # layers, incl. input and output.

s_ℓ : # nodes in layer ℓ

$$J(\alpha) = \left[\frac{1}{n} \sum_{k=1}^{n} J(\alpha; x_k, y_k) \right] + \frac{\lambda}{2} \sum_{\ell=1}^{n_\ell - 1} \sum_{i=1}^{s_\ell} \sum_{j=1}^{s_{\ell+1}} \left[\alpha_{ji}^{(\ell)} \right]^2$$

Quick notes on objective function and finding parameters

- Form is amenable to classification, just one-hot encode the output and use classification error rate as your objective
- For regression, be sure to scale the output variables to lie in the range of the activation function.
- Solving the objective function involves a form of gradient search
  - The partial derivatives are found via a technique called backpropogation

Tensorflow playground

On **this website** you'll find a cool interactive tool that allows you to play with NN for classification.

1. What are the hyperparameters of the model? Can you explain what each one does?
2. Try fitting the "exclusive or" (choose on top left) data set.
3. Also try fitting the "Spiral" data set.
4. Possible spiral solution:

Tensorflow playground

On **this website** you'll find a cool interactive tool that allows you to play with NN for classification.

1. What are the hyperparameters of the model? Can you explain what each one does?
2. Try fitting the "exclusive or" (choose on top left) data set.
3. Also try fitting the "Spiral" data set.
4. Possible spiral solution:
    1. Learning rate 0.03
    2. Two hidden layers, six and four neurons each
    3. Tanh activation
    4. Include all but $X_1 X_2$ features.
    5. L1 regularization, regularization rate $= 0.001$

Tensorflow playground

On **this website** you'll find a cool interactive tool that allows you to play with NN for classification.

1. What are the hyperparameters of the model? Can you explain what each one does?
2. Try fitting the "exclusive or" (choose on top left) data set.
3. Also try fitting the "Spiral" data set.
4. Possible spiral solution:
    1. Learning rate 0.03
    2. Two hidden layers, six and four neurons each
    3. Tanh activation
    4. Include all but $X_1 X_2$ features.
    5. L1 regularization, regularization rate $= 0.001$
5. You got close by trial and errror. What's another way?

Tensorflow playground

On **this website** you'll find a cool interactive tool that allows you to play with NN for classification.

1. What are the hyperparameters of the model? Can you explain what each one does?
2. Try fitting the "exclusive or" (choose on top left) data set.
3. Also try fitting the "Spiral" data set.
4. Possible spiral solution:
    1. Learning rate 0.03
    2. Two hidden layers, six and four neurons each
    3. Tanh activation
    4. Include all but $X_1 X_2$ features.
    5. L1 regularization, regularization rate $= 0.001$
5. You got close by trial and errror. What's another way?
    - Cross validation! Grid search, randomized search
    - But everything is computationally intense.

What's going on in the hidden layers?

Hover over the hidden layers in the tensorflow playground.

Q: What are we looking at?

What's going on in the hidden layers?

Hover over the hidden layers in the tensorflow playground.

Q: What are we looking at?

Ans: The scalar output of that neuron's activation function at each point in the feature space.