Neural Network representation

Michał Szafraniuk

michal.szafraniuk@gmail.com

September 11, 2019

1 Intro

This is a note elaborating on `deeplearning.ai` Coursera MOOC notation for basic neural network. The aim is to show a notation used from a bit more math rigorous perspective. It covers as forward prop as an leading example.

1.1 Notation

Let's consider a fully-connected neural network with input layer and L noninput layers. The example of such network with $L = 2$ is depicted on Figure 1.

Let's use the following notation:

- $l \in \{0, 1, ..., L\}$ for indexing layers,
- $n^{[l]}$ for the number of units in layer l,
- m for the number of examples,
- $g^{[l]} : \mathbb{R} \to \mathbb{R}$ (or g for brevity) for activation function or one of its lifted versions,
- $z_i^{[l]} \in \mathbb{R}$ for activation argument and $a_i^{[l]} \in \mathbb{R}$ for activation value for unit i in layer l (we will abuse notation by using this also to refer to the unit itself) for some unspecified example,
- $z_i^{[l](k)} \in \mathbb{R}$ and $a_i^{[l](k)} \in \mathbb{R}$: same as above but for specified, kth example,
- $z^{[l]} \in \mathbb{R}^{n^{[l]}}$ for activation argument vector for layer l and $a^{[l]} \in \mathbb{R}^{n^{[l]}}$ for activation value vector for that layer l (we will abuse notation by using this also to refer to the layer itself), unspecified example,
- $z^{[l](k)} \in \mathbb{R}^{n^{[l]}}$ and $a^{[l](k)} \in \mathbb{R}^{n^{[l]}}$: same as above but for specified, kth example,
- $Z^{[l]} \in \mathbb{R}^{n^{[l]} \times m}$ and $A^{[l]} \in \mathbb{R}^{n^{[l]} \times m}$ for m-batched matrix versions.

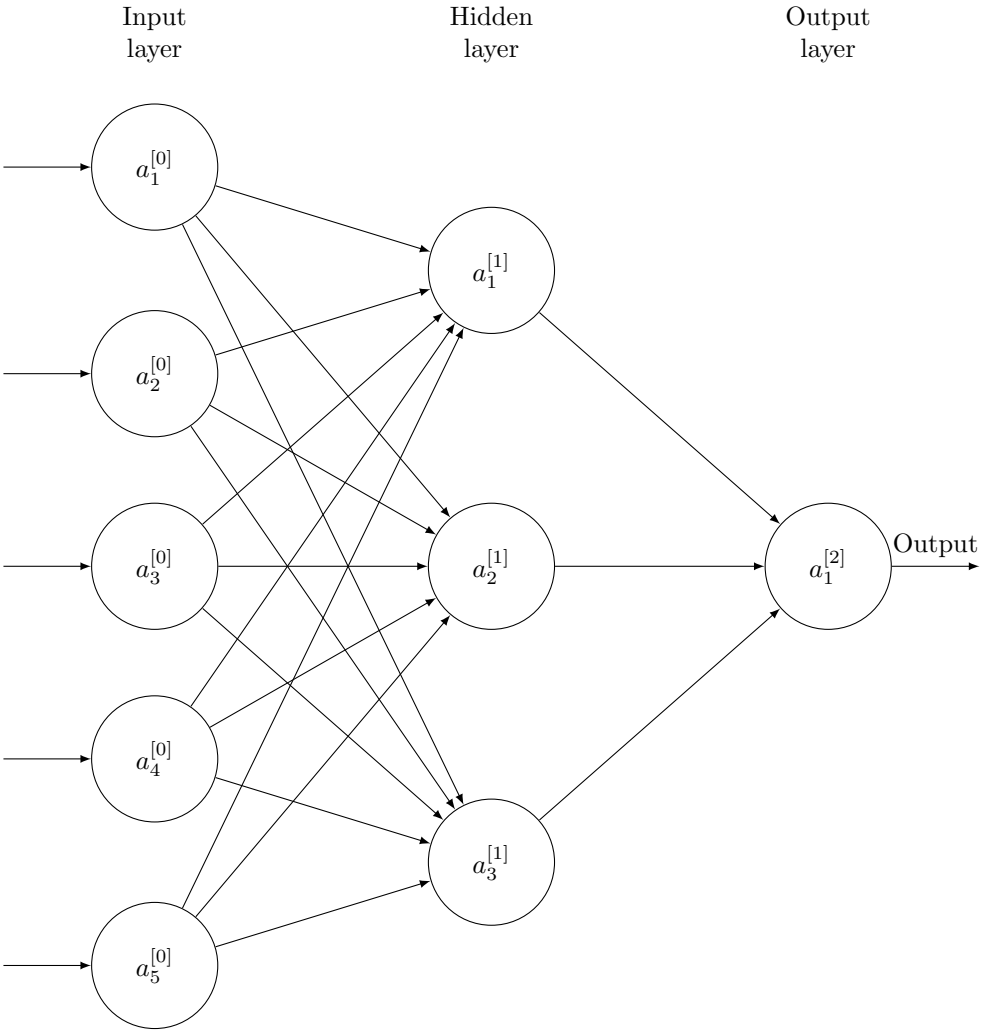

Figure 1: Fully-connected shallow network. Credit goes to Darren Reading.

1.2 Architecture

The general network architecture is as follows:

- one input layer ($l = 0$),
- $L - 1$ hidden layers ($l = 1, ..., L - 1$), each with $n^{[l]}$ units,
- one output layer ($l = L$).

1.3 Bit more details

1.3.1 Input layer

The input layer consists of units representing the features of an input example. Suppose each of our input examples have $n \equiv n^{[0]}$ features. Each example can be represented by:

$$\boldsymbol{x}^{(i)} = \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{bmatrix} = a^{[0]} \in \mathbb{R}^n$$

Please mind we call it $a^{[0]}$ for notation simplicity and consistency.

1.3.2 Hidden layers

Each hidden layer l has $n^{[l]}$ units. As we consider fully-connected network, each such unit have exactly $n^{[l-1]}$ edges (connections) coming in and $n^{[l+1]}$ coming out. So between each two consequtive layers, say $l - 1$ and l there is a bundle of $n^{[l-1]}n^{[l]}$ in-between connections. Each such connection has an associated parameter that determines the importance of that connection in overall computation.

1.3.3 Parameters

Training a network is all about tweaking parameters. This is one way how you can think of them:

- Each unit has its own scalar parameter $b_i^{[l]}$. These parameters form a column vector $b^{[l]} \in \mathbb{R}^{n^{[l]}}$ for the whole layer l.
- Each connection between single unit j in layer $l - 1$ and single unit i in layer l has its own scalar parameter $w_{i,j}^{[l]}$

  $$a_j^{[l-1]} \xrightarrow{w_{i,j}^{[l]}} a_i^{[l]}$$

  These parameters form o column vector $w_i^{[l]} \in \mathbb{R}^{n^{[l-1]}}$ representing connection parameters associated with unit i (i.e. going out of all units in layer $l - 1$ and coming into unit i in layer l)

  $$\langle a_j^{[l-1]} \rangle_{j=1..n^{[l-1]}} \equiv a^{[l-1]} \xrightarrow{w_i^{[l]}} a_i^{[l]}$$

There are $n^{[l]}$ such vectors, each for one unit in layer l. But again these column vectors can be stacked horizontally to form a matrix $W^{[l]} \in \mathbb{R}^{n^{[l]} \times n^{[l-1]}}$ that sits between the layers $l-1$ and l:

$$a^{[l-1]} \xrightarrow{W^{[l]}} a^{[l]}$$

1.3.4 Forward prop computations

In each unit there is a two-step baby computation happening in forward-prop phase:

- first, linear combination of all inputs from the previous layer and associated parameters is computed
- secondly, the result is applied to a function activatation

The final result is the output that goes from this particular unit to all the units in the next layer (as input to the same but next layer computation).

1.4 Perspectives

You can take 3 perspectives when thinking about computations in the network:

1. single unit (nonbatch) POV,
2. single layer / nonbatch (one training example) POV,
3. single layer / batch (more than one example) POV.

Let's now take a bit more formal look at the process from those different perspectives.

2 Nonbatch perspective

2.1 Single unit POV

Let's now consider a single unit i from hidden layer l. Using the notation above we have $i \in \{1, ..., n^{[l]}\}$ and $l \in \{1, ..., L\}$.

The first baby-step computation goes as follows:

$$z_i^{[l]} = \sum_{j=1}^{n^{[l-1]}} a_j^{[l-1]} w_{i,j}^{[l]} + b_i[l]$$

All the objects here are just scalars: we sum up products of all incoming activation inputs from previous layer with their associated parameters and add an ith unit corresponding b parameter. We can rewrite the above equation with a bit more vectorized notation:

$$z_i^{[l]} = w_i^{[l]T} a^{[l-1]} + b_i[l]$$

where $z_i^{[l]}$ and $b_i^{[l]}$ are still scalars but we arranged the l-layer connection parameters and outputs from layer $l-1$ into column vectors, i.e. $w_i^{[l]}, a^{[l-1]} \in \mathbb{R}^{n^{[l-1]}}$.

The second baby-step computation is just the application of activation function $g : \mathbb{R} \to \mathbb{R}$ to get the final activation value outgoing from unit i:

$$a_i^{[l]} = g(z_i^{[l]})$$

2.2 Single layer/nonbatch POV

We can generalize to a layer perspective to see how a computation for a single example works.

First, we can stack together the scalars $z_i^{[l]}$ and $b_i^{[l]}$ for $i \in \{1, ..., n^{[l]}\}$ to form column, non-batch, layer-associated vectors $z^{[l]}$ and $b^{[l]}$, both being objects from $\mathbb{R}^{n^{[l]}}$. Secondly, we need to generalize the single unit associated vector $w_i^{[l]}$ into layer associated matrix $W^{[l]} \in \mathbb{R}^{n^{[l]} \times n^{[l-1]}}$. But we already did that so we arrive at the first teen-step computation:

$$z^{[l]} = W^{[l]} a^{[l-1]} + b[l]$$

The second step is just lifting up the activation function to more vectorized version $g^{[l]} : \mathbb{R}^{n^{[l]}} \to \mathbb{R}^{n^{[l]}}$ in the following way:

$$a^{[l]} = g^{[l]}(z^{[l]})$$

3 Batch perspective

So far we have adopted a framework for a single example computation. But we can easily adapt this framework to an m-batch by further vectorizing.

All we need to do is to stack vertically $z^{[l](k)}$ and $a^{[l](k)}$ (both from $\mathbb{R}^{n^{[l]}}$) for $k \in \{1, ..., m\}$ to form layers-associated matrices $Z^{[l]}$ and $A^{[l]}$ (again both from $\mathbb{R}^{n^{[l]} \times m}$), broadcast $b^{[l]}$ up to $\mathbb{R}^{n^{[l]} \times m}$ and rewrite the forward prop equations to even more vectorized form:

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]}$$

$$A^{[l]} = g^{[l]}(Z^{[l]})$$

These are now adult-steps you make between each two, consequtive layers in for loops.

4 Summary

symbol	space	description
$g, g^{[l]}$	$\mathbb{R} \to \mathbb{R}$ or $\mathbb{R}^{n^{[l]}} \to \mathbb{R}^{n^{[l]}}$	activation function or its lifted versions
$z_i^{[l]}$	$\mathbb{R}$	activation argument for unit i in layer l for unspecified example
$z_i^{[l](k)}$	$\mathbb{R}$	activation argument for unit i in layer l for kth example
$a_i^{[l]}$	$\mathbb{R}$	activation value for unit i in layer l for unspecified example (or the unit itself)
$a_i^{[l](k)}$	$\mathbb{R}$	activation value for unit i in layer l for kth example (or the unit itself)
$z^{[l]}$	$\mathbb{R}^{n^{[l]}}$	activation argument vector for layer l
$a^{[l]}$	$\mathbb{R}^{n^{[l]}}$	activation value vector for layer l for unspecified example
$Z^{[l]}$	$\mathbb{R}^{n^{[l]} \times m}$	m-batched matrix of activation argument for layer l
$A^{[l]}$	$\mathbb{R}^{n^{[l]} \times m}$	m-batched matrix of activation value for layer l
$b_i^{[l]}$	$\mathbb{R}$	unit-associated parameter
$b^{[l]}$	$\mathbb{R}^{n^{[l]}}$ or $\mathbb{R}^{n^{[l]} \times m}$	vector of unit-associated parameters for layer l or its m-batch broadcast version
$w_{i,j}^{[l]}$	$\mathbb{R}$	parameter associated with connection between single unit j in layer $l-1$ and single unit i in layer l
$w_i^{[l]}$	$\mathbb{R}^{n^{[l-1]}}$	olumn vector representing connection parameters associated with unit i in layer l
$W^{[l]}$	$\mathbb{R}^{n^{[l]} \times n^{[l-1]}}$	matrix of connection parameters associated with layer l