<https://www.coursera.org/learn/machine-learning/discussions/weeks/4/threads/ml3GJjjbEeiwNxKOr03W_g>

Checking forward propagation understanding

I had to make some corrections, so that if someone see this in the future, they can understand something, or just get more confused, because I'm not sure if it is correct or not ^^.Considering:

* m = training examples
* n = features
* k = classes
* L = last layer
* If a network has Sj units in layer j, Sj+1 units in layer j+1, then *θ*(j) will be of dimension Sj+1 x Sj+1

Take care when looking at subscripts and transposes.

We have:

*X*=*a*(1), where *a*(1)∈*R*(*m* *x* *n*+1), with bias node.

Thus, *a*(1) rows are the input examples and the columns are the inputs’ features.

*θ*(1)∈*R*(*Sj*+1 *x* *S1+1*) = *θ*(1)∈*R*(*S1*+1 *x* *S1+1*) = *θ*(1)∈*R*(*S2* *x* *S1+1*)

So, ***θ*(1)** rows are the weights (parameters) of each node connection from S1+1 layer to the S2 layer and the columns are the values of each parameter. In the case of ***θ*(1)** (first parameter matrix) columns are the features of each training example, so that *S1+*1 *and n*+1 are equal. Therefore, ***θ*′(1)** is the transpose of that.

Calculating:

*a*(1)∗*θ*′(1)=*z*(2), where *z*(2)∈*R*(*m* *x* *S­2*)

Where *z*(2) matrix units’ are values used to help to compute the final layer, so they don’t have much meaningful sense.

Using activation function:

*a*(2)=*σ*(*z*(2)), where *a*(2)∈*R*(*m* *x* *S­2*)

Now we have *a*(2) matrix units’ are *z*(2) which were activated using the sigmoid function, but still they don’t have much meaningful sense.

Next, we add the bias node to *a*(2), getting *a*(2)∈*R*(*m* *x* *S2+1*)

Then, we start repeating the process:

*θ*(2)∈*R*(*S2+1* *x* *S2+1*) = *θ*(2)∈*R*(*S3* *x* *S2+1*)

*a*(2)∗*θ*′(2)=*z*(3), where *z*(3)∈*R*(*m* *x* *S3*)

…

When we reach *a*(*L*), which is the last layer, we will have *a*(*L*)∈*R*(*m* *x* *Sj*), where rows represent examples and the columns represent the classes, in the last layer Sj = K. Therefore, each unit of this matrix is the probability from 0 to 1 of a example *m* belonging to a class *k*. Consequently, we have to find the maximum probability in each row, then take its index. This index will represent which class our neural network predicted that *m* example to belong. For example, if the highest value of the row is in index 1 (position 2), that *m* example probably belongs to class *k*=2.

Is there any problem with my reasoning?