50.039 – Theory and Practice of Deep learning

Alex

Week 09

[The following notes are compiled from various sources such as textbooks, lecture materials, Web resources and are shared for academic purposes only, intended for use by students registered for a specific course. In the interest of brevity, every source is not cited. The compiler of these notes gratefully acknowledges all such sources.]

1 Task 1:

Understanding how the gradient flows ...

Refer to the neural networks in figures 1 and 2 with inputs $x_1 \in \mathbb{R}^D$ to neuron n_1 and $x_2 \in \mathbb{R}^D$ to neuron n_2. Equation for any neuron is

$$n_i = g(b + \sum_k w_{k,i} n_k + \sum_l v_{l,i} x_l), \tag{1}$$

where $w_{k,1} = 0, w_{k,2} = 0 \ \forall k$ (for n_1 and n_2) and $v_{l,i} = 0 \ \forall l$ for all neurons except n_1 and n_2. Note also that for many neurons n_i several $w_{k,i}$ are zero, whenever n_k is no input to n_i, for example in figure 1 $w_{4,3}$ and $w_{4,2}$ are zero.

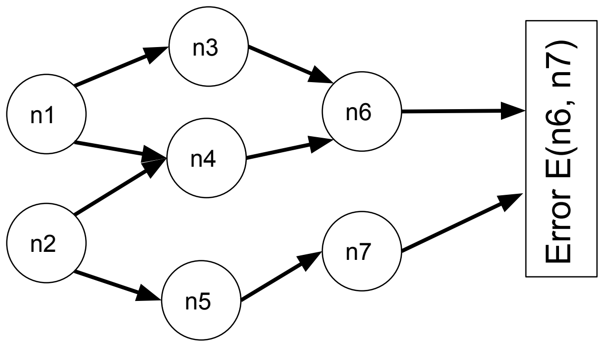

Figure 1: Mini Neural Networks I for Task 1

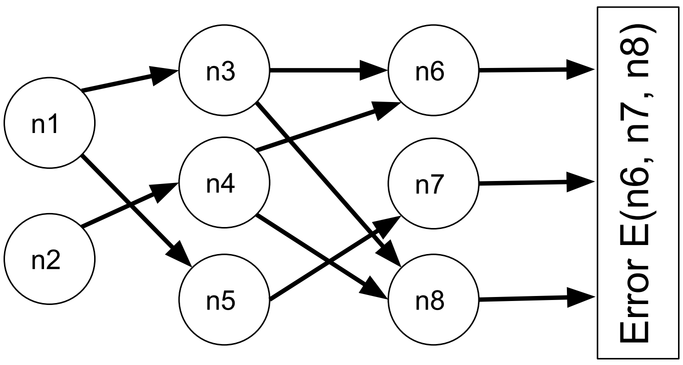

Figure 2: Mini Neural Networks II for Task 1

We assume here for simplicity of partial derivatives, that the outputs of neurons which enter the loss function E are one-dimensional.
For the error E in both neural networks, instead of using the softmax function we learned in class, we use the quadratic error function for regression purpose,

$$E = \sum_{i \in data} (n_6 - y_6^{(i)})^2 + (n_7 - y_7^{(i)})^2$$

$$E = \sum_{i \in data} (n_6 - y_6^{(i)})^2 + (n_7 - y_7^{(i)})^2 + (n_8 - y_8^{(i)})^2$$

Note that the output neurons depend implicitly on the neural network inputs $x_1^{(i)}$ and $x_2^{(i)}$
Write down an expression for the gradients of all the weights for E function for each neural network: For figure 1:

1. $\frac{\partial E}{\partial n_4}$;
2. $\frac{\partial E}{\partial w_{2,5}}$;
3. $\frac{\partial E}{\partial (v_{1,1})_d}$ where $(v_{1,1})_d$ is the d-th dimension of the weight for neuron n_1 for input x_1;
4. $\frac{\partial E}{\partial (x_2)_d}$ where $(x_2)_d$ is he d-th dimension of the input for neuron n_2;

For figure 2:

1. $\frac{\partial E}{\partial (v_{2,2})_d}$ where $(v_{2,2})_d$ is the d-th dimension of the weight for neuron n_2 for input x_2;
2. $\frac{\partial E}{\partial w_{2,4}}$
3. $\frac{\partial E}{\partial n_1}$

note: Write the expression in terms of

- $\frac{\partial E}{\partial n_i}$, where n_i is a neuron directly connected to the output

- $\frac{\partial n_k}{\partial n_i}$, where n_i is direct input to n_k
- $\frac{\partial n_i}{\partial w_{k,i}}$
- $\frac{\partial n_i}{\partial v_{i,i}}$
- and $\frac{\partial n_k}{\partial x_k}$ if x_k is input to the neural network (otherwise use $\frac{\partial n_k}{\partial n_i}$)
- At this point you do **not need** to plugin how $\frac{\partial n_k}{\partial n_i}$ or $\frac{\partial n_k}{\partial w_k}$ or $\frac{\partial n_k}{\partial x_k}$ looks like.
- You **do not need** to multiply out terms in parentheses, so $(a+b)c$ or $((a+b)c+(d+e)f)g$ is fine to keep it like that!

2 Task 2:

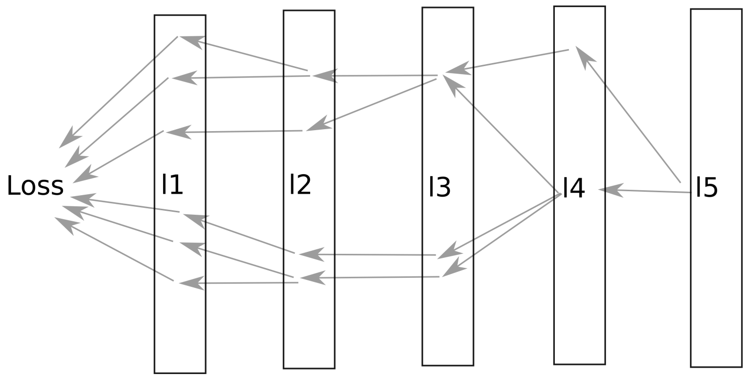

Figure 3: paths from a neuron in layer 5 to the loss function

Consider figure 3. Try to understand that a neuron in layer 5 may have many different paths by which is contributes to the loss function (depicted in light gray). The gradient during backpropagation will flow along each of these paths from the loss to that nameless neuron in layer 5 .
If you would compute the gradient $\frac{\partial E}{\partial n_k}$ for a fixed neuron in layer k as in Task 1 (with the terms of Task 1), then it would be a sum of terms, one term for each path from the loss function E to that neuron in layer $k > 1$. If you are not sure about this, then look at your solutions for task 1.
Consider now one of these paths from the loss function E to a neuron in layer $k > 1$.

How many terms would be in the product – which corresponds to a single path from the loss function E to a neuron in layer $k > 1$? The relevant parameter is the layer index k.