TTIC 31230 Fundamentals of Deep Learning
Midterm Preparation Material

We will provide with the exam copies of all the particular pages of particular lecture slides listed below. Other than that the exam will be closed book. The midterm covers the following topics (The material for lectures 8 and 9 should be added tonight).

Lecture 2: The basics of the education framework edf (slides 38 through 44). The edf implementation edf.py of ps1.

You should be able to implement a simple computation node class similar to Sigmoid or Affine as implemented in edf.py of ps1.

Lecture 3: Index notation (slides 6 through 9). The basic index notation definitions of convolutional layers (slides 23, 25, 28 and 29).

You should be able write similar explicit index equations defining slight modifications of CNNs.

Example: Consider a log-frequency spectrogram $x[b,t,f,c]$ where b is the batch index, t is a time index, and f is a frequency index, and c is a channel index. For example, you should be able to write the explicit index equation for a $T \times F$ convolution over both time and frequency with stride in time of s_t and a stride in frequency of s_f. (Two such CNN layers are the modern first layers in speech recognition).

Example: Suppose that we want to implement scale invariance in computer vision. The input image x (with RGB channels) is downsampled (or smoothed) so that we have a "pyramid" of images at different scales — a sequence of images where each image has half the pixel dimensions of the previous image but where this is done by simple smoothing. We will work with a system of images $x[s,d,i,j,c]$ where s is a scale index, d is a depth index, i and j are spacial indeces and c is a channel index (here we omit the batch index). We let $x[0,0,i,j,c]$ be the input image and let $x[s,0,i,j,c]$ be the input image smoothed to scale s defined by

$$x[s+1,0,i,j,c] = x[s,0,2i+\Delta i, 2j\Delta j, c]$$

where Δi and Δj range over $\{0,1\}$. Here $x[s+1,0,i,j,c]$ is an RGB (three channel) image and $x[s+1,0,i,j,c]$ has half the spatial dimensions of $x[s,0,i,j,c]$.

We compute $x[s+1,d+1,i,j,c]$ from $x[s+1,d,i,j,c]$ by a convolution with filter W^d. Write this convolution equation defining $x[s+1,d+1,i,j,c]$ assuming that all convolutions are done with 4×4 filters at stride 2.

Assume that the input dimensions are $2^n \times 2^n$ and that we stop doing convolutions when the spacial dimension reaches 1×1. The final image representation is the concatenation of the top level channel vectors from all input scales.

What are the spatial dimension of $x[s,d,i,j,c]$?

Write a convolution equation where $x[s+1,d+1,i,j,c]$ is computed from both $x[s,d+1,i,j,c]$ and $x[s+1,d,i,j,c]$ as the sum of two convolutions. Assume another system of filters U^d for the dependence of $x[s{+}1,d,i,j,c]$ on $x[s,d,i,j,c]$.

Lecture 4. Xavier Initialization (slide 23) Batch Normalization (slides 27 through 32), Resnet (slides 35 through 39)

Example: Rewrite the spatial batch normalization equations for one dimensional convolution on a time signal $x[b,t,c]$ where b is the batch index, t is a time index, and c is the channel index.

Lecture 5. High level highway architectures (skip connections) equations (slide 6), the update gate RNN (UGRNN) architecture architecture (slide 10). Language modeling (slides 12 through 15). Machine Translation (slides 16 through 19). Attention (slides 32 and 33).

Example: Write the equations for a gated RNN where the "diversion" d^t is the same as in the UGRNN but where both a forget gate and an input gate are used as in an LSTM.

Example: Convert bits per character to word level perplexity in a language where the words average nine characters (not including the space between words).

Example: rewrite the last two equations on slide 33 to handle the case of image captioning where the attention for generating word $s+1$ is over the pixels (i,j) of an image $x[i,j,c]$. Note that $x[i,j]$ is a vector.

Lecture 6. Classical convergence theorem (slide 9). The theoretical relationship between learning rate and batch size (slide 13). The nonstandard momentum formulation (slide 25) predicted to make the learning rate independent of momentum and the standard one (slide 26) for which we predict the learning rate goes as $\eta_0(1-\mu)$. RMSprop and Adam (slides 28 and 29).

Example: Suppose the total object has a unique point as its optimum. Argue that in this case SGD cannot will not converge to that point unless the learning rate goes to zero.

Example: Suppose $\eta_t > 0$, that $\sum_{t=0}^{\infty} \eta_t$ is finite. Further suppose that the total loss $E_{x,y} \sim$ Train is quadratic with a unique minima. Give conditions under which SGD will not converge to the minimum.

Example: Based on the expected interaction between learning rate and momentum, what might one expect the interaction to be between the β_1 parameter of Adam and the learning rate?

Example: Consider logistic regression defined by $\mathrm{loss}(x,y) = -\log_2 Q_w(y|x)$ and $Q_W(y|x) = \mathrm{softmax}_y\, Wx$ where x is a vector and W is a matrix. If components of x have units of meters and the weights in W have units reciprocal meters (value per meter), and the loss function has units of bits, what must the units of the learning rate be to make the SGD update equation have correct units.

Example: If we scale the inputs by $x' = sx$ for some scaling factor s (possibly corresponding to a change in units) and inverse scale the initial weights $W' = (1/s)W$ should we change the learning rate of simple (Vanilla) SGD, and if so how?

Example: Repeat the above two examples but for RMSProp rather than simple SGD.

Lecture 7. L_2 regularization and Dropout (slides 13 through 17). The free lunch theorem (slides 24, 26 and 27). The KL divergence bound (slide 33). The L_2 bound (slide 34). The Implicit prior bound (slide 40).

Example: When using L_2 regularization an optimum of the objective is a compromise between training error and norm of the parameter vector. State the relationhip that holds between the total gradient g and the parameter vector Φ at the minimum of the regularized objective.

$$\Phi^* = \mathrm{argmin} + \Phi \mathrm{loss}(\mathrm{Train}) + \frac{1}{2}\lambda||\Phi||^2$$

Example: Repeat the above calculation but for L_1 regularization.

$$\Phi^* = \mathrm{argmin} + \Phi \mathrm{loss}(\mathrm{Train}) + \lambda||\Phi||_1$$

$$||\Phi||_1 = \sum_i |\Phi_i|$$

Example: Drive the L_2 generalization bound from the KL-divergence generalization bound by showing that for

$$P(\Psi) = \frac{1}{Z}e^{-\frac{1}{2}||\Psi||^2}$$

and

$$Q(\Psi) = \frac{1}{Z}e^{-\frac{1}{2}||\Psi-\Phi||^2}$$

we have

$$KL(Q,P) = \frac{1}{2}||\Phi||^2.$$

Hint: Define $\epsilon \doteq \Psi - \Phi$ and write the KL-divergence as

$$KL(Q,P) = E_Q\ f(\epsilon, \Phi + \epsilon)$$

where we have $E_Q\epsilon = 0$.

Lecture 8. Entropy and the source coding theorem (slides 9 through 12). Log loss as cross entropy (slides 18 and 19). KL-divergence (slides 20 to 22).

Fundamental inequalities (slide 23) and the fact that these inequalities prove that no code can use fewer bits than the entropy and that the optimum of log loss is the population distribution. Differential entropy (slides 30-33). Mutual Information (slides 36 through 38).

Example: Show $H(P) \geq 0$. (very easy).

Example: Give an example of a continuous density p for which $H(p) < 0$ as measured by differential entropy.

Example: Derive the equivalence of the four expressions for mutual information on slides 36 and 38 starting from only the definitions

$$\begin{aligned}
H(x) &\doteq E_{x\sim\mathrm{Pop}} - \log \mathrm{Pop}(x) \\
H(y|x) &\doteq E_{x,y\sim\mathrm{Pop}} - \log \mathrm{Pop}(y|x) \\
I(x,y) &\doteq KL(\mathrm{Pop}(x,y), \mathrm{POP}(x) \times \mathrm{Pop}(y)) \\
\\
&= E_{x,y\sim\mathrm{Pop}}\ \log \frac{\mathrm{Pop}(x,y)}{\mathrm{Pop}(x)\mathrm{Pop}(y)}
\end{aligned}$$

Example: derive the fundamental inequalities on slide 23 starting only form the above definitions.

Lecture 9. Basic Definitions (slides 4 through 8). The expression for f.grad (slide 11). The fact that in tree structured models everything can be done efficiently and that message passing (loopy BP) is often applied successfully to non-tree (loopy) models. That MCMC is impractical. Pseudolikelihood (slides 26 and 27). Contrastive Divergence (slide 30).

Example: In a graphical model with N nodes, and V possible values per node, what is time complexity of computing $\tilde{Q}(\hat{y})$ for the pseudolikelihood $\tilde{Q}$ from $\hat{y}$ and the graphical model tensor f?

Example: Consider contrastive divergence on a graphical model N nodes, and V possible values per node and where contrastive divergence is based on Gibbs sampling where we select a node and random and replace it with a random draw from the conditional distribution defined by its neighbors. What is the running time doing a CD gradient update defined by iterating over all nodes and computing a gradient value from a single Gibbs update to that node and then summing these gradient updates for the SGD step.