Exam Advanced Machine Learning
22 October 2020, 15.30–18.15

This exam consists of 6 problems, each consisting of several questions. All answers should be motivated, including calculations, formulas used, etc. The use of a calculator is not allowed.

Question 1: Short questions
Please provide an argument for your answer on the following questions.

(a) Is the following statement true of false? Stochastic gradient descent, even with small step size, sometimes increases the loss in some iteration for convex problems.

SGD due to stochasticity does not necessarily decrease the loss in each iteration. One can construct case where there could be a data point which is sort of contradicting with all others, so optimizing this particular data point may increase the overall loss.

(b) You observe the following train and test error as a function of model complexity p for three different models. Consider the minimum possible bias for each model over all settings of p for $0 \leq p \leq 30$. Compare the minimum bias of the three models (i.e., are they the same, which one has the highest, which one has the lowest)?

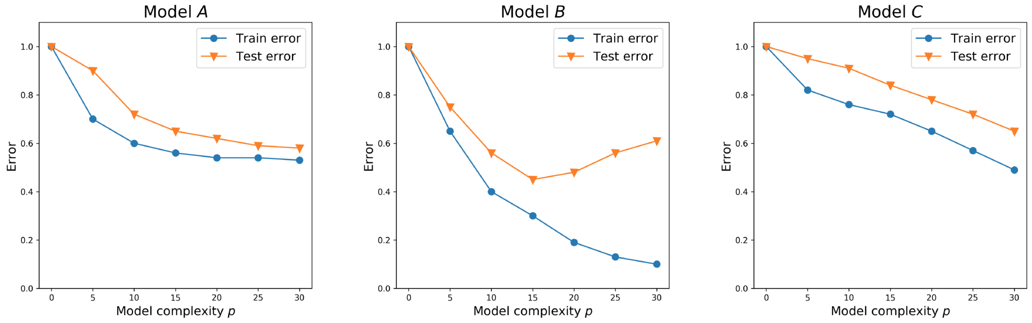

Figure 1: Train and test error.

The train error is a good indicator of the bias. B has the lowest possible among the three models, and A and C have around the same.

(c) This question still applies to the models for which the train and test error is depicted in Figure 1. For which values of p and for which models does the test and train error indicate overfitting, if any? Also, which models, if any, appear to be underfit for all settings of p?

A divergence in the test and train error indicates overfitting. This only happens for model B at $p = 20$ and $p = 30$. The test and train error are still decreasing for model C, so it is possible it is underfit.

(d) Is the following statement true of false? For logistic regression, with parameters optimized using a stochastic gradient method, setting parameters to 0 is an acceptable initialization.

True, it is a convex problem.

(e) Recall the LSTM architecture (see Figure 2). Suppose you want the memory cell to sum its inputs over time. What values should you fix in the LSTM cell to achieve this, and what values would you choose?

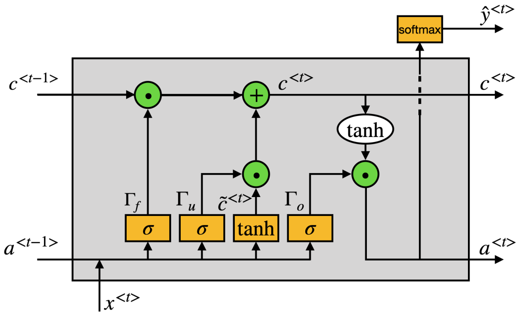

Figure 2: The LSTM cell.

Set input = 1, forget = 1.

(f) Alice and Barbara are trying to redesign the LeNet convolutional network architecture to reduce the number of weights. Alice wants to reduce the number of feature maps in the first convolution layer. Barbara wants to reduce the number of hidden units in the last layer before the output. Explain which approach is better.

Barbara's approach is better because most of the weights are in the fully connected layers of LeNet (or a typical conv net architecture).

Question 2: Neural networks

Consider the following convolutional neural network architecture.

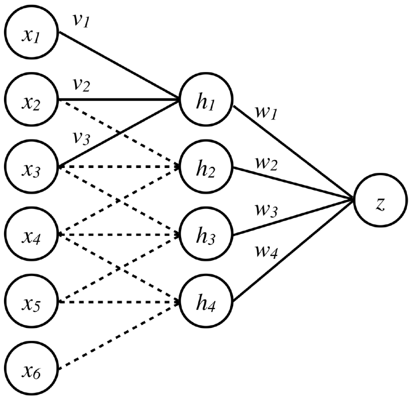

Figure 3: One-dimensional convolutional neural network.

In the first layer, we have a one-dimensional convolution with a single filter of size 3 such that $h_i = \sigma\left(\sum_{j=1}^{3} v_j x_{i+j-1}\right)$. The second layer is fully connected, such that $z = \sum_{i=1}^{4} w_i h_i$.

The hidden units' activation function $\sigma(x)$ is the logistic (sigmoid) function. The output unit is linear (no activation function). We perform gradient descent on the loss function $E = (y - z)^2$, where y is the training label for $x = (x_1, \dots, x_6)$.

(a) What is the total number of parameters in this neural network? Recall that convolutional layers share weights. There are no bias terms.

The answer is 7. There are 3 parameters in layer 1 and 4 parameters in layer 2.

(b) Compute $\partial E / \partial w_i$.

$$\frac{\partial E}{\partial w_i} = -2(y - z)h_i.$$

(c) Compute $\partial E / \partial v_j$.

$$\frac{\partial E}{\partial v_j} = -2(y-z)\frac{\partial z}{\partial v_j} = -2(y-z)\sum_{i=1}^{4}\frac{\partial z}{\partial h_i}\frac{\partial h_i}{\partial v_j} = -2(y-z)\sum_{i=1}^{4} w_i h_i (1 - h_i) x_{i+j-1}.$$

(d) One of the difficulties with the logistic activation function is that of saturated units, which prohibits learning. Briefly explain the problem, and whether switching to a $\tanh$ activation function fixes the problem.

No, switching to $\tanh$ does not fix the problem. The derivative of $\sigma(z)$ is small for large negative or positive z. The same problem persists in $\tanh(z)$. Both functions have a sigmoidal shape. We can see that $\tanh$ is effectively a scaled and translated sigmoid function: $\tanh(z) = 2\sigma(2z) - 1$.

Question 3: Graphical models

The following figure shows a graphical model over six binary-valued variables $A, \dots, F$. We do not know the parameters of the probability distribution associated with the graph.

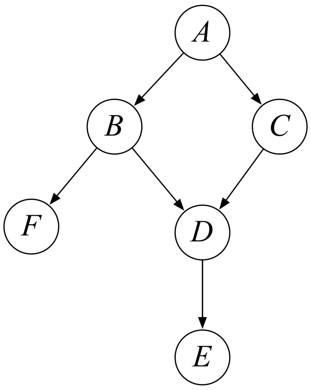

Figure 4: Graphical model.

(a) Write the expression for the joint probability $\mathbb{P}(A, B, C, D, E, F)$ of the network in its *reduced* factored form.

$$\mathbb{P}(A)\mathbb{P}(B|A)\mathbb{P}(C|A)\mathbb{P}(D|B, C)\mathbb{P}(E|D)\mathbb{P}(F|B).$$

(b) Which of the following conditional independence assertions are true?

i) $A \perp\!\!\!\perp E$
ii) $B \perp\!\!\!\perp C \,|\, A$
iii) $F \perp\!\!\!\perp C \,|\, A$
iv) $B \perp\!\!\!\perp C \,|\, A, E$

i) False
ii) True
iii) True
iv) True

Question 4: Hidden Markov Models (HMMs)

Consider a six-state hidden Markov model specified by (π, A, φ) that can output 4 possible values. Thus, the hidden states $z_i \in \{1, \dots, 6\}$, and the output values $x_i \in \{a, b, c, d\}$. The further specification of the hidden Markov model is given as follows:

$$\pi = (1, 0, 0, 0, 0, 0), \quad A = \begin{pmatrix} 0 & 0.3 & 0 & 0.7 & 0 & 0 \\ 0 & 0 & 0.8 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad \varphi = \begin{pmatrix} 0.5 & 0.3 & 0 & 0.2 \\ 0.1 & 0.3 & 0.5 & 0.1 \\ 0.2 & 0.3 & 0.4 & 0.1 \\ 0.3 & 0.2 & 0.3 & 0.2 \\ 0 & 0.3 & 0.2 & 0.6 \\ 0.4 & 0.4 & 0.1 & 0.1 \end{pmatrix}.$$

Thus, $\mathbb{P}(z_{t+1} = 3|z_t = 2) = 0.8$, and $\mathbb{P}(z_{t+1} = 4|z_t = 1) = 0.7$. But also, $\mathbb{P}(x_t = c|z_t = 2) = 0.5$, and $\mathbb{P}(x_t = a|z_t = 4) = 0.3$. We will use the following shorthanded notation where we write $\mathbb{P}(x = abca, z_2 = 1, z_4 = 2)$ instead of $\mathbb{P}(x_1 = a, x_2 = b, x_3 = c, x_4 = a, z_2 = 1, z_4 = 2)$. For each of the items below, insert $<$, $>$, or $=$ in the square brackets between the left and the right expression. Justify your answer. Hint: thinking before computing might save a lot of time.

(a) $\mathbb{P}(x = abca, z_1 = 1, z_2 = 2) \quad [\quad] \quad \mathbb{P}(x = abca|z_1 = 1, z_2 = 2)$.

$<$. The right hand side is the left hand side divided by $\mathbb{P}(z_1 = 1, z_2 = 2)$ which is less than 1.

(b) $\mathbb{P}(x = abca, z_1 = 1, z_4 = 6) \quad [\quad] \quad \mathbb{P}(x = abca|z_1 = 1, z_4 = 6)$.

$=$. Here, $\mathbb{P}(z_1 = 1, z_4 = 6) = 1$, hence we have equality.

(c) $\mathbb{P}(x = acdb, z_2 = 2, z_3 = 3) \quad [\quad] \quad \mathbb{P}(x = acdb, z_2 = 4, z_3 = 5)$.

$<$. We work out $\mathbb{P}(x = acdb, z_2 = 2, z_3 = 3) = C\mathbb{P}(z_2 = 2|z_1 = 1)\mathbb{P}(z_3 = 3|z_2 = 2)\mathbb{P}(x_2 = c|z_2 = 2)\mathbb{P}(x_3 = d|z_3 = 3) = C \times 0.3 \times 0.8 \times 0.5 \times 0.1$. This is less than $\mathbb{P}(x = acdb, z_2 = 4, z_3 = 5) = C\mathbb{P}(z_2 = 4|z_1 = 1)\mathbb{P}(z_3 = 5|z_2 = 2)\mathbb{P}(x_2 = c|z_2 = 2)\mathbb{P}(x_3 = d|z_3 = 3) = C \times 0.7 \times 1 \times 0.3 \times 0.6$ for some constant C.

(d) $\mathbb{P}(x = acdb)$ [ ] $\mathbb{P}(x = acdb|z_2 = 4, z_3 = 5)$.

$<$. We work out the probabilities $\mathbb{P}(x = acdb, z_2 = 2, z_3 = 3)$, $\mathbb{P}(x = acdb, z_2 = 2, z_3 = 5)$, $\mathbb{P}(x = acdb, z_2 = 4, z_3 = 5)$, and sum them to get $\mathbb{P}(x = acdb)$. To get $\mathbb{P}(x = acdb|z_2 = 4, z_3 = 5)$, we can divide $\mathbb{P}(x = acdb, z_2 = 4, z_3 = 5)$by $\mathbb{P}(z_2 = 4, z_3 = 5) = 0.7$.

(e) Describe the differences between a Hidden Markov Model and a more general Bayesian Network.

An HMM is a time series model, where each random variable has at most one parent. It has a specific structure, determined by the parameterization according the prior, transition, and observation distributions. An arbitrary BN can be any acyclic graph, along with the associated CPTs.

Question 5: Reinforcement learning

Consider the grid-world given below and Pacman who is trying to learn the optimal policy. If an action results in landing into one of the shaded states the corresponding reward is awarded during that transition. All shaded states are terminal states, i.e., the process terminates once arrived in a shaded state. The other states have the *North, East, South, West* actions available, which deterministically move Pacman to the corresponding neighboring state (or have Pacman stay in place if the action tries to move out of the grid). Assume the discount factor $\gamma = 0.5$ and the Q-learning rate $\alpha = 0.5$ for all calculations. Pacman starts in state (1, 3).

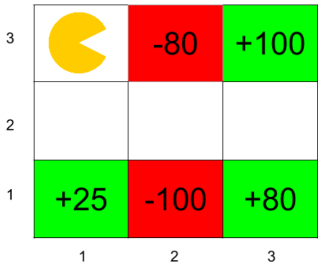

Figure 5: Pacman strikes back!

(a) What is the value of the optimal value function V^* at the following states: $V^*(3, 2)$, $V^*(2, 2)$, and $V^*(1, 3)$? Recall that $V^*(s)$ represents the expected return under the optimal policy starting in state s.

$V^*(3, 2) = 100$, $V^*(2, 2) = 50$, and $V^*(1, 3) = 12.5$. The optimal values for the states can be found by computing the expected reward for the agent acting optimally from that state onwards. Note that you get a reward when you transition *into* the shaded states and not *out* of them. So for example the optimal path starting from (2,2) is to go to the +100 square which has a discounted reward of $0 + \gamma * 100 = 50$. For (1,3), going to either of +25 or +100 has the same discounted reward of 12.5.

(b) The agent starts from the top left corner and you are given the following episodes from runs of the agent through this grid-world. Each line in an episode is a tuple containing (s, a, s', r).

Using Q-Learning updates, what are the following Q-values after the above three episodes: $Q((3, 2), N)$, $Q((1, 2), S)$, and $Q((2, 2), E)$?

Episode 1	Episode 2	Episode 3
(1,3), S, (1,2), 0	(1,3), S, (1,2), 0	(1,3), S, (1,2), 0
(1,2), E, (2,2), 0	(1,2), E, (2,2), 0	(1,2), E, (2,2), 0
(2,2), S, (2,1), -100	(2,2), E, (3,2), 0	(2,2), E, (3,2), 0
	(3,2), N, (3,3), +100	(3,2), S, (3,1), +80

$Q((3,2),N) = 50$, $Q((1,2),S) = 0$, and $Q((2,2),E) = 12.5$. Q-values obtained by Q-learning updates: $Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha(R(s,a,s') + \gamma \max_{a'} Q(s',a'))$

(c) Consider a feature-based representation of the Q-value function:

$$Q_f(s,a) = w_1 f_1(s) + w_2 f_2(s) + w_3 f_3(a).$$

Here, $f_1(s)$ represents the x-coordinate of the state, and $f_2(s)$ represents the y-coordinate of the state, and $f_3(N) = 1$, $f_3(S) = 2$, $f_3(E) = 3$, $f_3(W) = 4$. Given that all w_i are initially 0, what are their values after the first episode? Note that the weight updates are given by $w_i \leftarrow w_i + \alpha[(r(s,a,s') + \gamma \max_{a'} Q(s',a')) - Q(s,a)]f_i(s,a)$.

$w_1 = -100$, $w_2 = -100$, and $W_3 = -100$. Using the approximate Q-learning weight updates: $w_i \leftarrow w_i + \alpha[(R(s,a,s') + \gamma \max_{a'} Q(s',a')) - Q(s,a)]f_i(s,a)$. The only time the reward is non zero in the first episode is when it transitions into the -100 state.

(d) Assume the weight vector w is equal to $(1,1,1)$. What is the action prescribed by the Q-function in state $(2,2)$?

The action prescribed at (2,2) is $\max_a Q((2,2),a)$ where $Q(s,a)$ is computed using the feature representation. In this case, the Q-value for *West* is maximum $(2+2+4=8)$.

Question 6: The Adam optimizer

In this question, we are going to look at the Adam optimizer in more detail. Recall that, given the gradient g_t calculated at epoch t, the Adam optimizer has three distinct steps. First, update the moving averages by $v_t = \beta_1 v_{t-1} + (1-\beta_1)g_t$ and $s_t = \beta_2 s_{t-1} + (1-\beta_2)g_t^2$. Second, apply the bias correction $v_t^c = v_t/(1-\beta_1^t)$ and $s_t^c = s_t/(1-\beta_2^t)$. Third, update the parameters by $w_{t+1} = w_t - \alpha v_t^c/(\sqrt{s_t^c} + \varepsilon)$.

(a) Show that s_t can be expressed only in terms of the gradients $g_1, \dots, g_t$ by the expression $s_t = (1-\beta_2)\sum_{i=1}^t \beta_2^{t-i} g_i^2$.

Since $s_t = \beta_2 s_{t-1} + (1-\beta_2)g_t^2$, this implies that $s_{t-1} = \beta_2 s_{t-2} + (1-\beta_2)g_{t-1}^2$, and so on. Therefore, replacing s_{t-1} in the first equation, gives us $s_t = \beta_2(\beta_2 s_{t-2} + (1-\beta_2)g_{t-1}^2) + (1-\beta_2)g_t^2$. Simplifying yields $s_t = \beta_2^2 s_{t-2} + (1-\beta_2)(g_t^2 + \beta_2 g_{t-1}^2)$. Therefore, $s_t = (1-\beta_2)(g_t^2 + \beta_2 g_{t-1}^2 + \dots + \beta_2^{t-1} g_1^2)$. Or equivalently, $s_t = (1-\beta_2)\sum_{i=1}^t \beta_2^{t-i} g_i^2$.

(b) Given the expression of s_t in part (a), what is $\mathbb{E}[s_t]$ in terms of $\mathbb{E}[g_t^2]$ and β_2? You may assume that the g_i's are independent and identically distributed.

Starting from $s_t = (1-\beta_2)\sum_{i=1}^t \beta_2^{t-i} g_i^2$, we have $\mathbb{E}[s_t] = \mathbb{E}[(1-\beta_2)\sum_{i=1}^t \beta_2^{t-i} g_i^2]$. This leads to $\mathbb{E}[s_t] = (1-\beta_2)\sum_{i=1}^t \beta_2^{t-i}\mathbb{E}[g_i^2]$. Using the i.i.d. assumption, we have $\mathbb{E}[s_t] = \mathbb{E}[g_t^2](1-\beta_2)\sum_{i=1}^t \beta_2^{t-i}$. This computes to $\mathbb{E}[s_t] = \mathbb{E}[g_t^2](1-\beta_2)(\beta_2^t - 1)/(\beta_2 - 1)$. And thus, $\mathbb{E}[s_t] = \mathbb{E}[g_t^2](1-\beta_2^t)$. This is the motivation behind the bias correction step.

(c) The result that you obtained in part (b) explains why you do the bias correction step. Using your result in the previous part, explain what would happen if you did not perform the bias correction step.

In the long term – both expectations would converge. However, initially, the value is biased toward zero. This bias is worse with larger values of β_2.

partial grade	1	2	3	4	5	6
(a)	1	1	1	1	1	1
(b)	2	2	3	1	2	2
(c)	2	2		1	2	2
(d)	1	1		2	1	
(e)	2			1		
(f)	1					

Final grade is: (sum of partial grades) / 4.0 + 1.0