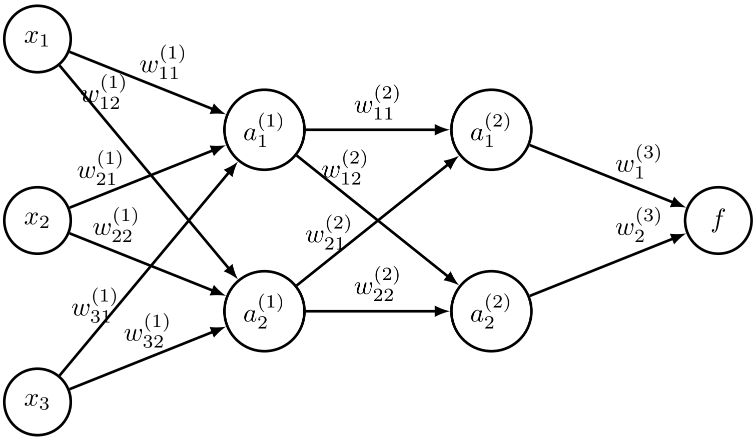
Exercises
Introduction to Machine Learning
SS 2020

Institute for Machine Learning
Dept. of Computer Science, ETH Zürich
Prof. Dr. Andreas Krause
Web: `https://las.inf.ethz.ch/teaching/introml-s20`
Email questions to:
Shubhangi Ghosh, shghosh@student.ethz.ch,

Series 4, April 6th, 2020 (Neural Networks)

Problem 1 (Dropout and Back Propagation):

Xiaoming designed the following neural network to predict his grades in exams. He bases the predictions on his degree of being nervous (x_1), his mood(x_2) and the weather on the exam day (x_3). In the hidden layers, he uses the sigmoid function $\sigma(x) = 1/(1+e^{-x})$. In the output layer, no activation function is used. As in many other regression tasks, he uses $\mathcal{L}_2$ as the loss function: $L = (y - f)^2$



Xiaoming collected one training example x_1, x_2, x_3 with his grade y in Introduction to Machine Learning exam. Help him to write down the sequence of calculations and compute the loss by answering the following questions.

1. what is $a_i^{(1)}$?

   (a) $\frac{1}{1+exp(-\Sigma w_{ki}x_k)}$

   (b) $\frac{1}{1+exp(\Sigma w_{ki}x_k)}$

   (c) $\sigma(\Sigma w_{ki}x_k)$

   (d) $\sigma(-\Sigma w_{ki}x_k)$

   Solution:
   (a) and (c) are both correct solutions.

2. what is $a_i^{(2)}$?

   (a) $\frac{1}{1+exp(-\Sigma w_{ki}\alpha_k)}$

   (b) $\frac{1}{1+exp(\Sigma w_{ki}\alpha_k)}$

   (c) $\sigma(\Sigma w_{ki}\alpha_k)$

(d) $\sigma(-\Sigma w_{ki}\alpha_k)$

Solution:
(a) and (c) are both correct solutions.

3. what is f ?

(a) $w_1^{(3)}\alpha_1^{(2)} + w_2^{(3)}\alpha_2^{(2)}$

(b) $w_1^{(3)}\alpha_1^{(2)}w_2^{(3)}\alpha_2^{(2)}$

(c) $\frac{1}{1+exp(-\Sigma w_i^{(3)}\alpha_i^{(2)})}$

(d) $\frac{1}{1+exp(\Sigma w_i^{(3)}\alpha_i^{(2)})}$

Solution:
(a) is the correct solution.

After some semesters of attending exams, Xiaoming finds out he can not collect enough training samples. So he decides to use a dropout technique to reduce overfitting of his model. In particular, he applies dropout for the 2nd hidden layer ($a_1^{(2)}$ and $a_2^{(2)}$) with the probability of the corresponding neuron being retained being 0.4. Help him compute the expected value of the loss in this case, given training example x_1, x_2, x_3 and grade y, by answering the following questions.

4. What is $\mathbb{E}_{a_1^{(2)},a_2^{(2)}}(f)$?

(a) $0.4(w_1^{(3)}\alpha_1^{(2)} + w_2^{(3)}\alpha_2^{(2)})$

(b) $0.6(w_1^{(3)}\alpha_1^{(2)} + w_2^{(3)}\alpha_2^{(2)})$

(c) $0.4w_1^{(3)}\alpha_1^{(2)} + 0.6w_2^{(3)}\alpha_2^{(2)}$

(d) $0.6w_1^{(3)}\alpha_1^{(2)} + 0.4w_2^{(3)}\alpha_2^{(2)}$

Solution:
(a) is the correct solution.

5. What is $Var_{a_1^{(2)},a_2^{(2)}}(f)$?

(a) $0.24((w_1^{(3)}\alpha_1^{(2)})^2 + (w_2^{(3)}\alpha_2^{(2)})^2)$

(b) $0.24(w_1^{(3)}\alpha_1^{(2)} + w_2^{(3)}\alpha_2^{(2)})$

(c) $0.16((w_1^{(3)}\alpha_1^{(2)})^2 + (w_2^{(3)}\alpha_2^{(2)})^2)$

(d) $0.16(w_1^{(3)}\alpha_1^{(2)})^2 + 0.64(w_2^{(3)}\alpha_2^{(2)})^2$

Solution:
(a) is the correct solution.

6. What is $\mathbb{E}(L)$?

(a) $Y^2 - 2Y\mathbb{E}(f) + \mathbb{E}(f^2)$

(b) $Y^2 - 2Y\mathbb{E}(f) + Var(f) + (\mathbb{E}(f))^2$

(c) $Y^2 + 2Y\mathbb{E}(f) + \mathbb{E}(f^2)$

(d) $Y^2 + 2Y\mathbb{E}(f) + Var(f) + (\mathbb{E}(f))^2$

Solution:
(a) and (b) are both correct solutions.

At a certain iteration of training, Xiaoming inputs his training example x_1, x_2, x_3 and grade y, and looks into his neural network after the forward pass. He finds that $a_1^{(2)}$ gets dropped out and $a_2^{(2)}$ is kept. To perform a SGD update to weight $w_{21}^{(1)}$, Xiaoming:

- Executes forward pass according to answers to (1), (2) and (3) while setting $\alpha_1^{(2)}$ to zero.
- Runs backward pass computing the derivative, $\frac{dL}{dw_{21}^{(1)}}$ according to the above.
- Update parameter according to SGD rule.

7. Help him compute the derivative, $\frac{dL}{dw_{21}^{(1)}}$.

(a) $2(f-y)w_2^{(3)}\phi'(w_{21}^{(2)}\alpha_1^{(1)} + w_{22}^{(2)}\alpha_2^{(1)})w_{12}^{(2)}\phi'(w_{11}^{(1)}x_1 + w_{21}^{(1)}x_2 + w_{31}^{(1)}x_3)x_2$

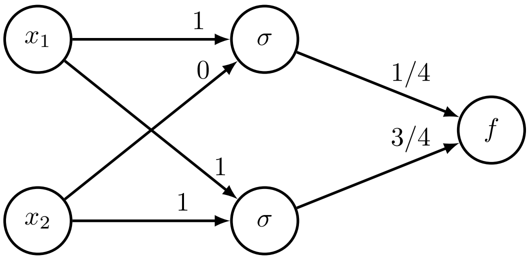
(b) $2(f-y)w_2^{(3)}\phi'(w_{12}^{(2)}\alpha_1^{(1)} + w_{22}^{(2)}\alpha_2^{(1)})w_{12}^{(2)}\phi'(w_{12}^{(1)}x_1 + w_{22}^{(1)}x_2 + w_{32}^{(1)}x_3)x_2$

(c) $2(f-y)w_2^{(3)}\phi'(w_{12}^{(2)}\alpha_1^{(1)} + w_{22}^{(2)}\alpha_2^{(1)})w_{12}^{(2)}\phi'(w_{11}^{(1)}x_1 + w_{21}^{(1)}x_2 + w_{31}^{(1)}x_3)x_2$

Solution:
(c) is the correct solution.

Problem 2 (2018 Exam Question: ANN):

Consider the neural network given in the figure below. The numbers above the lines correspond to the weights of the connections. In the hidden layer, the activation function σ is applied and the network does not have any biases.



Read off the initial weights $\mathbf{W}_1$ and $\mathbf{w}_2$ from the network given above, such that the weights correspond to the following matrix notation of the neural network with input $\mathbf{x} = (x_1, x_2)$.

$$f(\mathbf{x}; \mathbf{W}_1, \mathbf{w}_2) = \mathbf{w}_2^\top \sigma(\mathbf{W}_1 \mathbf{x})$$

1. What is $\mathbf{W}_1$?

   (a) $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$

   (b) $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$

   (c) $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$

   (d) $\begin{pmatrix} 1 & 0 \\ \frac{1}{4} & \frac{3}{4} \end{pmatrix}$

   Solution:
   (b) is the correct solution.

2. What is $\mathbf{w}_2$?

   (a) $\begin{pmatrix} 1 \\ \frac{3}{4} \end{pmatrix}$

   (b) $\begin{pmatrix} 1 \\ \frac{1}{4} \end{pmatrix}$

   (c) $\begin{pmatrix} \frac{3}{4} \\ \frac{1}{4} \end{pmatrix}$

   (d) $\begin{pmatrix} \frac{1}{4} \\ \frac{3}{4} \end{pmatrix}$

   Solution:
   (d) is the correct solution.

3. You are given a data set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ with n data points, where $\mathbf{x}_i \in \mathbb{R}^2$ and $y_i \in \mathbb{R}$ for $i = 1, \ldots, n$. Calculate the empirical risk $\hat{R}(\mathcal{D}, \mathbf{W}_1, \mathbf{w}_2)$ for training the neural network using the squared loss with added L_2 regularization *only* on the output layer.

(a) $\frac{1}{n}\Sigma_{i=1}^{n}(\mathbf{w}_2^T\mathbf{W}_1 x_i - y_i)^2 + \|\mathbf{w}_2\|^2$

(b) $\frac{1}{n}\Sigma_{i=1}^{n}(\mathbf{w}_2^T\sigma(\mathbf{W}_1 x_i) - y_i)^2 + \|\mathbf{w}_2\|^2$

(c) $\frac{1}{n}\Sigma_{i=1}^{n}(\mathbf{w}_2^T\sigma(\mathbf{W}_1 x_i) - y_i) + \|\mathbf{w}_2\|^2$

(d) $\frac{1}{n}\Sigma_{i=1}^{n}(\mathbf{W}_1^T\sigma(\mathbf{w}_2 x_i) - y_i)^2 + \|\mathbf{w}_2\|^2$

Solution:
(b) is the correct solution.

4. For training the neural network, the empirical risk function is often minimized using stochastic gradient descent (SGD). SGD and standard gradient descent have different computational complexities. What is the computational complexity of SGD?

   Assume, n is the number of data points.

   (a) $\mathcal{O}(1)$ per gradient step

   (b) $\mathcal{O}(n)$ per gradient step

   (c) $\mathcal{O}(n^2)$ per gradient step

   (d) $\mathcal{O}(log(n))$ per gradient step

   Solution:
   (a) is the correct solution.

5. What is the computational complexity of standard gradient descent?

   (a) $\mathcal{O}(1)$ per gradient step

   (b) $\mathcal{O}(n)$ per gradient step

   (c) $\mathcal{O}(n^2)$ per gradient step

   (d) $\mathcal{O}(log(n))$ per gradient step

   Solution:
   (b) is the correct solution.

Given $\mathbf{W}_1$ and $\mathbf{w}_2$ as in the diagram on the previous page, find different weights $\tilde{\mathbf{W}}_1$ and $\tilde{\mathbf{w}}_2$, such that the resulting network has the same performance as the original network *(for any activation function σ)*. In other words, find $\tilde{\mathbf{W}}_1$ and $\tilde{\mathbf{w}}_2$ s.t. $f(\mathbf{x}; \mathbf{W}_1, \mathbf{w}_2) = f(\mathbf{x}; \tilde{\mathbf{W}}_1, \tilde{\mathbf{w}}_2)$, but $\mathbf{W}_1 \neq \tilde{\mathbf{W}}_1$ and $\mathbf{w}_2 \neq \tilde{\mathbf{w}}_2$.

6. What is $\tilde{\mathbf{W}}_1$?

   (a) $\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$

   (b) $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$

   (c) $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$

   (d) $\begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$

   Solution:
   (d) is the correct solution.

7. What is $\tilde{\mathbf{w}}_2$?

   (a) $\begin{pmatrix} \frac{3}{4} \\ \frac{3}{4} \end{pmatrix}$

   (b) $\begin{pmatrix} \frac{1}{4} \\ \frac{3}{4} \end{pmatrix}$

   (c) $\begin{pmatrix} \frac{3}{4} \\ \frac{1}{4} \end{pmatrix}$

   (d) $\begin{pmatrix} \frac{1}{4} \\ \frac{1}{4} \end{pmatrix}$

   Solution:
   (c) is the correct solution.

8. In this question, we show that if σ is the *relu* activation function $\sigma(x) = \max(0, x)$, then the resulting network $f(\mathbf{x}; \mathbf{W}_1, \mathbf{w}_2)$ is a piecewise linear function in $\mathbf{x}$. Match the 4 intervals on the real line to the resulting linear function in the interval.

   (a) $x_1 \leq 0$ and $x_1 + x_2 \leq 0$
   (b) $x_1 \geq 0$ and $x_1 + x_2 \geq 0$
   (c) $x_1 \geq 0$ and $x_1 + x_2 \leq 0$
   (d) $x_1 \leq 0$ and $x_1 + x_2 \geq 0$

   □ $f(x) = \frac{x_1}{4}$
   • $f(x) = 0$
   ∗ $f(x) = \frac{3(x_1+x_2)}{4}$
   – $f(x) = x + \frac{3x_2}{4}$

   Solution:
   (a) and •
   (c) and □
   (d) and ∗
   (b) and –

Problem 3 (Expressiveness of Neural Networks):

In this question we will consider neural networks with sigmoid activation functions of the form

$$\varphi(z) = \frac{1}{1 + \exp(-z)}.$$

If we denote by v_j^l the value of neuron j at layer l its value is computed as

$$v_j^l = \varphi\left(w_0 + \sum_{i \in \mathsf{Layer}_{l-1}} w_{j,i} v_i^{l-1}\right).$$

In the following questions you will have to design neural networks that compute functions of two Boolean inputs X_1 and X_2. Given that the outputs of the sigmoid units are real numbers $Y \in (0, 1)$, we will treat the final output as Boolean by considering it as 1 if greater than equal to 0.5 and 0 otherwise.

1. Give 3 weights w_0, w_1, w_2 for a single unit with two inputs X_1 and X_2 that implements the logical OR function $Y = X_1 \vee X_2$. Please note that w_0, w_1 and w_2 can only take values -0.5, 0, or 1.

   (a) $w_0 = ?$

   (b) $w_1 = ?$

   (c) $w_2 = ?$

   Solution:
   We consider the following network $w = -0.5$, $w_1 = 1$ and $w_2 = 1$. We check whether the output we get is desired OR function. The network looks as follows,

$$A \vee B = \mathrm{round}(\varphi(w_0 + w_1 A + w_2 B))$$

$$A \vee B = \mathrm{round}(\varphi(-0.5 + A + B))$$

A	B	$A \vee B$	Network	Round
1	1	1	≈ 0.81	1
0	1	1	≈ 0.62	1
1	0	1	≈ 0.62	1
0	0	0	≈ 0.37	0

2. Can you implement the logical AND function $Y = X_1 \wedge X_2$ using a single unit? If so, give weights that achieve this. If not, set the weights to 0.

   // Please note that w_0, w_1 and w_2 can only take values -2, -1.5, -1, -0.5, 0, 0.5 or 1.

   (a) $w_0 = ?$

   (b) $w_1 = ?$

   (c) $w_2 = ?$

   Solution:
   $w_0 = -1.5$, $w_1 = 1$ and $w_2 = 1$. We check whether the output we get is desired AND function. The network looks as follows,

$$A \wedge B = \mathrm{round}(\varphi(w_0 + w_1 A + w_2 B))$$

$$A \wedge B = \mathrm{round}(\varphi(-1.5 + A + B))$$

A	B	$A \wedge B$	Network	Round
1	1	1	≈ 0.62	1
0	1	0	≈ 0.37	0
1	0	0	≈ 0.37	0
0	0	0	≈ 0.18	0

3. It is impossible to implement the XOR function $Y = X_1 \oplus X_2$ using a single unit. However, you can do it using a multi-layer neural network. Use the smallest number of units you can to implement XOR function. Draw your network and show all the weights. What is the smallest number of hidden layers or hidden states to implement the XOR function?

   Solution:
   We find the weights by choosing the weights of the first layer and optimizing over the weights of the last layer s.t. the inequalities are satisfied.

We use a network with one hidden layer and two states

$$A \oplus B = \mathrm{round}(\varphi(-\varphi(A+B) + 0.84\varphi(2A+2B))$$

A	B	$A \oplus B$	Network	Round
0	0	0	0.480010659844	0
0	1	1	0.502202727467	1
1	0	1	0.502202727467	1
1	1	0	0.486027265451	0

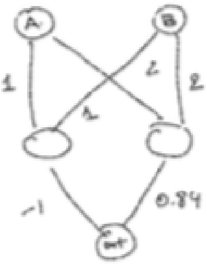


Figure 1

For sketch check Figure 1.