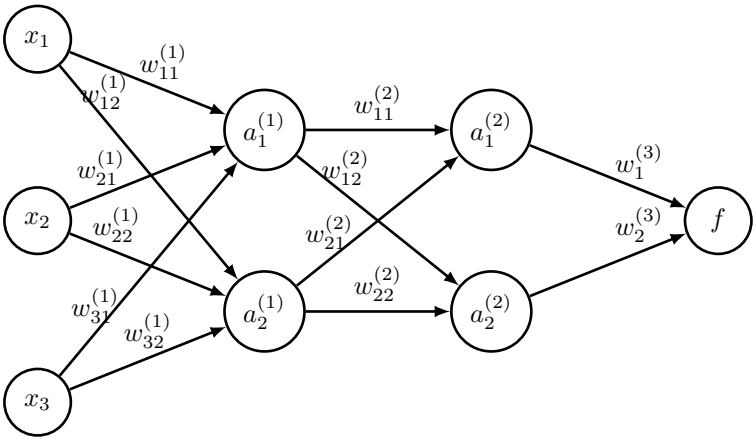
Exercises
Introduction to Machine Learning
SS 2019

Institute for Machine Learning
Dept. of Computer Science, ETH Zürich
Prof. Dr. Andreas Krause
Web: `https://las.inf.ethz.ch/teaching/introml-s19`
Email questions to:
jingwei.tang@inf.ethz.ch, kjong.lehmann@inf.ethz.ch

Series 4, April 1st, 2019 (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: $L = (y - f)^2$



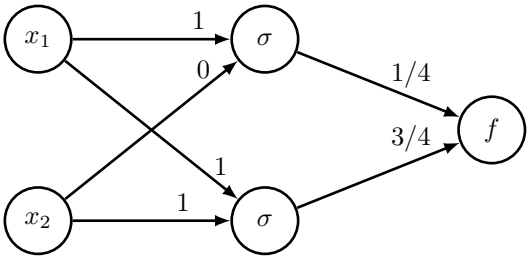
1. Xiaoming collected one training example x_1, x_2, x_3 with his grade y in Introduction to Machine Learning exam. Help him to write down the sequence of calculations and compute the loss.

2. 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 probability 0.4. Write down the expected value of the loss in this case, given training example x_1, x_2, x_3 and grade y.

3. 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. Help Xiaoming to perform a SGD update to weight $w_{21}^{(1)}$.

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.



1. 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 $\mathbf{x} = (x_1, x_2)$.

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

2. 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 $i = 1, \ldots, n$. Write down 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.

3. For training the neural network, the empirical risk function is often minimized using *stochastic* gradient descent (SGD). What is the difference between SGD and standard gradient descent in terms of computational complexity?

4. 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 $\mathbf{w}_2 \neq \tilde{\mathbf{w}}_2$.

5. 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 $\mathbf{x}$. Specifically, define 4 intervals on the real line where the resulting function is linear, for the weights given in the diagram on the previous page.

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

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

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 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 $Y = X_1 \vee X_2$.

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, explain the problem.

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.

4. Create a neural network with only one hidden layer (of any number of units) that implements

$$(A \vee \neg B) \oplus (\neg C \vee \neg D).$$

   Draw your network and show all the weights.