1. Find accuracy, precision, recall and F-score.

	Predicted class: Positive	Predicted class: Negative
actual class: Positive	85	15
actual class: Negative	890	10

SOL: accuracy: 95/1000, precision: 85/975, recall: 85/100,

TO BE GRADED: F-score: 2*precision*recall/(precision+recall)$\approx$ 0.158

2. Suppose we have binary classifier which uses $\sigma(x)$ for 1-dimensional input x and sigmoid function $\sigma(\cdot)$. The decision rule is such that, given threshold s, we decide input x is positive if $\sigma(x) > s$, and negative otherwise. Suppose the following 4 data samples are given in format (x_i, y_i) such that input x_i and output y_i where $y_i = 1$ represents that x_i is positive, and $y_i = 0$ means x_i is negative:

$$(-2, 0), (-1, 1), (1, 0), (2, 1)$$

Draw the ROC curve. What is AUC?
SOL: For $s = 0$, we achieve (FP rate, TP rate)=$(1, 1)$.

For $s = \sigma(-2)$, we achieve (FP rate, TP rate)=$(0.5, 1)$.

For $s = \sigma(-1)$, we achieve (FP rate, TP rate)=$(0.5, 0.5)$.

For $s = \sigma(1)$, we achieve (FP rate, TP rate)=$(0, 0.5)$.

For $s = 1.1$, we achieve (FP rate, TP rate)=$(0, 0)$.

AUC is 3/4.

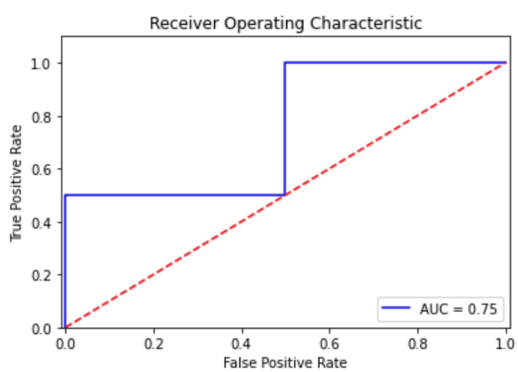

Figure 1: ROC curve

3. We learned about the information entropy of a discrete random variable (RV) X. Specifically, if X has probability mass function $p(x)$ such that $p(x) := P(X = x)$, its entropy $H(X)$ is given by

$$H(X) = -E\left[\log p(X)\right] = -\sum_x p(x)\log p(x)$$

Now, if X is *continuous* RV, and has probability density function $f(x)$, we can define its **differential entropy** $h(X)$ as follows:

$$h(X) = -E\left[\log f(X)\right] = -\int_{-\infty}^{\infty} f(x)\log f(x)dx$$

Note that the definitions are similar, but differential entropy has slightly different meaning from original entropy (it is related to the required # of bits to compress X, but some differences). Suppose X is Gaussian and $X \sim N(\mu, \sigma^2)$, that is

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Find differential entropy $h(X)$.

SOL:

$$\begin{aligned}
h(X) &= -\int f(x)\log f(x)\,dx \\
&= -\int f(x)\log\left[(2\pi\sigma^2)^{-1/2}\exp\left(-\tfrac{1}{2\sigma^2}(x-\mu)^2\right)\right]\,dx \\
&= \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\int f(x)(x-\mu)^2\,dx \\
&= \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\mathbb{E}[(x-\mu)^2] \\
&= \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2} = \frac{1}{2}\log(2\pi e\sigma^2)
\end{aligned}$$

4. Write down the cross-entropy loss L for the linear score for class 1, 2 and 3

$$s = \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix}$$

where the ground truth label for this data sample was class 2.
SOL:

$$L = -\log\left(\frac{\exp(1)}{\exp(2)+\exp(1)+\exp(0)}\right)$$

5. Consider the neural network in Fig. 2. The hidden layer uses ReLU activation. The output layer is a linear layer, and outputs the softmax of the linear score. Suppose the input is $(x_1, x_2) = (1, 3)$. What is the output (y_1, y_2)? Assume all the biases are 0.

SOL: TO BE GRADED: The output of the first neuron of the hidden layer is

$$\max(1\times(-2) + 3\times 1, 0) = 1$$

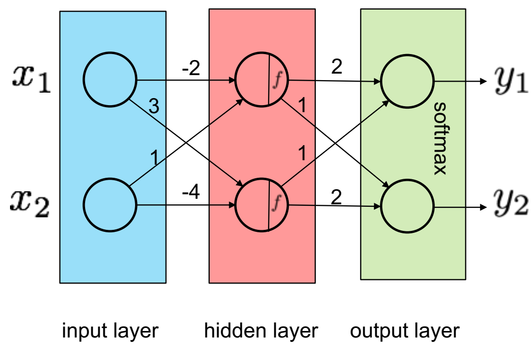

Figure 2: Neural Network

The output of the second neuron of the hidden layer is

$$\max(1 \times (3) + 3 \times (-4), 0) = 0$$

So the output of the hidden layer is $(1, 0)$. Then the output layer is

$$\text{softmax}(1 \times 2 + 0 \times 1, 1 \times 1 + 0 \times 2) = \text{softmax}(2, 1) = \left(\frac{\exp(2)}{\exp(2) + \exp(1)}, \frac{\exp(1)}{\exp(2) + \exp(1)} \right)$$

6. Consider loss function $L(x, y) = 2x + 3xy$ with learning rate α. We would like to minimize the loss using gradient descent. What is the step for gradient descent at point (x, y)?
   SOL: TO BE GRADED:

$$-\alpha \nabla L = -\alpha \begin{bmatrix} \frac{\partial L}{\partial x} \\ \\ \frac{\partial L}{\partial y} \end{bmatrix} = -\alpha \begin{bmatrix} 2 + 3y \\ 3x \end{bmatrix}$$

7. We have computational graph in Fig 3. The numbers above the flows represent the forward values. The numbers below the flows represent the gradient with respect to f. Fill in (a)–(d).

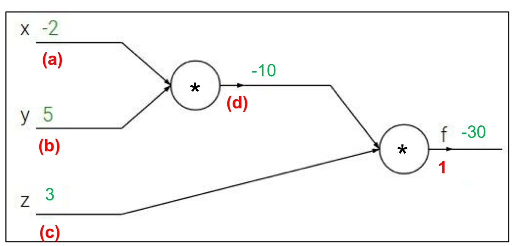

Figure 3: computational graph

SOL: (a):15, (b):-6, (c): -10, (d):3

8. Consider a simple linear classifier with m classes. The input to the classifier is vector $x \in \mathbb{R}^n$. The first layer is linear layer whose output is Wx, and $W \in \mathbb{R}^{m \times n}$ is a parameter matrix. Then the cross-entropy loss function denoted by L is applied to the output (that is, first applying softmax to the output and applying negative log-likelihood). Suppose that the current input is x, and the ground truth label is given by $y \in \{1, \ldots, m\}$. Find the expression for

$$\frac{dL}{dW}$$

Note that the answer should have the same shape as W.

SOL: Let us denote $\mathbf{e}_i$ as a one-hot vector in $\mathbb{R}^n$, that is, a vector with all zeros except i-th element which is 1. We have

$$\frac{dL}{dW} = (\text{softmax}(Wx) - \mathbf{e}_y)x^T$$