Exercise 2

Machine Learning in Graphics & Vision

Driton Goxhufi 4233242 driton.goxhufi@student.uni-tuebingen.de
Damir Ravlija 5503184 damir.ravlija@student.uni-tuebingen.de

1 Task 1

(a) Classification accuracy of the initialized model on the test dataset is 0.5

(b) Loss of the initialized model is 0.7149616252170096.

(c) In the first step of derivation we use the chain rule and the fact that $\frac{\partial f_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{x})}{\partial \boldsymbol{w}^T\boldsymbol{x}} = f_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{x})(1 - f_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{x})$ (1).

$$\begin{aligned}
\frac{\partial L(\boldsymbol{x}, t, \boldsymbol{w})}{\partial \boldsymbol{w}} &\overset{(1)}{=} \frac{1}{N}\sum_{n=1}^{N}\left[-t_n\frac{1}{f_{\boldsymbol{w}}(\boldsymbol{x}_n)}f_{\boldsymbol{w}}(\boldsymbol{x}_n)(1 - f_{\boldsymbol{w}}(\boldsymbol{x}_n))\boldsymbol{x}_n + (1 - t_n)\frac{1}{1 - f_{\boldsymbol{w}}(\boldsymbol{x_n})}f_{\boldsymbol{w}}(\boldsymbol{x}_n)(1 - f_{\boldsymbol{w}}(\boldsymbol{x}_n))\boldsymbol{x}_n\right] \\
&= \frac{1}{N}\sum_{n=1}^{N}\left[-t_n(1 - f_{\boldsymbol{w}}(\boldsymbol{x}_n))\boldsymbol{x}_n + (1 - t_n)f_{\boldsymbol{w}}(\boldsymbol{x}_n)\boldsymbol{x}_n\right] \\
&= \frac{1}{N}\sum_{n=1}^{N}\left[(-t_n + t_n f_{\boldsymbol{w}}(\boldsymbol{x}_n) + f_{\boldsymbol{w}}(\boldsymbol{x}_n) - t_n f_{\boldsymbol{w}}(\boldsymbol{x}_n))\boldsymbol{x}_n\right] \\
&= \frac{1}{N}\sum_{n=1}^{N}\left[f_{\boldsymbol{w}}(\boldsymbol{x}_n) - t_n\right]\boldsymbol{x}_n
\end{aligned}$$

(1)

$$\begin{aligned}
\frac{\partial f_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{x})}{\partial \boldsymbol{w}^T\boldsymbol{x}} &= \frac{\partial}{\partial \boldsymbol{w}^T\boldsymbol{x}}\left(\frac{1}{1 + e^{-\boldsymbol{w}^T\boldsymbol{x}}}\right) = \frac{e^{-\boldsymbol{w}^T\boldsymbol{x}}}{\left(1 + e^{-\boldsymbol{w}^T\boldsymbol{x}}\right)^2} = \frac{1 + e^{-\boldsymbol{w}^T\boldsymbol{x}} - 1}{\left(1 + e^{-\boldsymbol{w}^T\boldsymbol{x}}\right)^2} \\
&= \frac{1}{\left(1 + e^{-\boldsymbol{w}^T\boldsymbol{x}}\right)} - \frac{1}{\left(1 + e^{-\boldsymbol{w}^T\boldsymbol{x}}\right)^2} \\
&= \frac{1}{\left(1 + e^{-\boldsymbol{w}^T\boldsymbol{x}}\right)}\left(1 - \frac{1}{\left(1 + e^{-\boldsymbol{w}^T\boldsymbol{x}}\right)}\right) = f_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{x})(1 - f_{\boldsymbol{w}}(\boldsymbol{w}^T\boldsymbol{x}))
\end{aligned}$$

After $1,000$ iterations the loss and accuracy of the model are:

loss = 0.3868595564299156

accuracy = 0.83

(d) Since $f_{\boldsymbol{w}}$ represents the class posterior probability $p(c_1|\boldsymbol{x})$, the loss is minimal for $t = 1$ when this probability is the highest (i.e. $f_{\boldsymbol{w}} = 1$). As the confidence of the model decreases, the loss function grows up to infinity which corresponds to the case when the model is confident in the wrong class, i.e. assigns $f_{\boldsymbol{w}} = 0$ (see Figure (1).

On the other hand, if the label $t = 0$, the loss is minimized when $p(c_2|\boldsymbol{x}) = 1 - p(c_1|\boldsymbol{x}) = 1 - f_{\boldsymbol{w}}$ is maximized which is achieved when $f_{\boldsymbol{w}} = 0$. The loss is higher when the model assigns higher probability to the other class, and grows to infinity as $f_{\boldsymbol{w}}$ nears 1.

Driton Goxhufi 4233242
Damir Ravlija 5503184

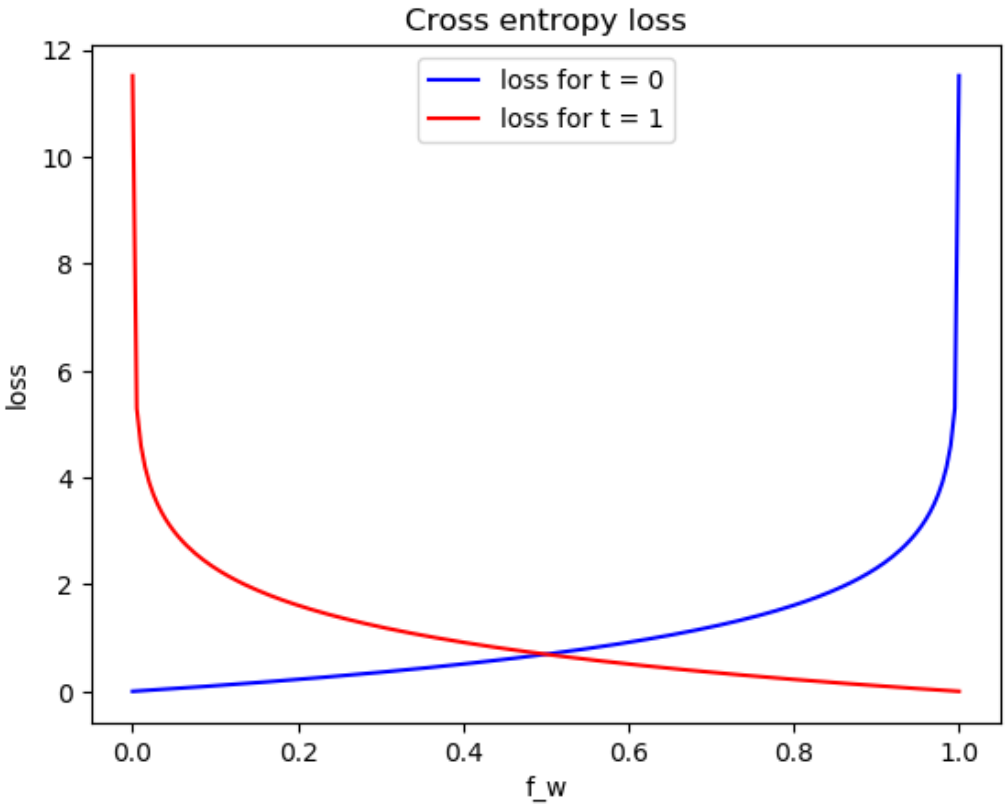


Figure 1: Plot of results from task (d)

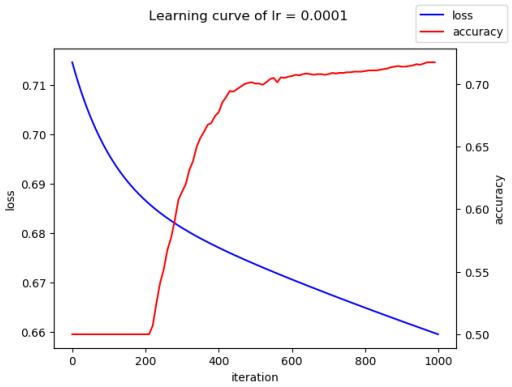
(e) We observe that the lower the learning rate is, the increase in accuracy and loss decrease are slower but more stable over time. Even after $1,000$ iterations, smaller learning rates show increasing accuracy and decreasing loss trends, which implies that they would benefit from more iterations. Optimal learning rate that shows the best combination of stability and convergence speed is 0.1. Both loss and accuracy achieve the best values when the learning rate is set to 0.1.

In the extreme case of learning rate set to 1, both loss and accuracy are extremely unstable. This instability probably arises from overly large steps that the gradient descent makes and which causes it to overstep the minima. Conversely, in case of the smallest learning rate 0.0001 the accuracy of the model starts increasing only after around 200 iterations.

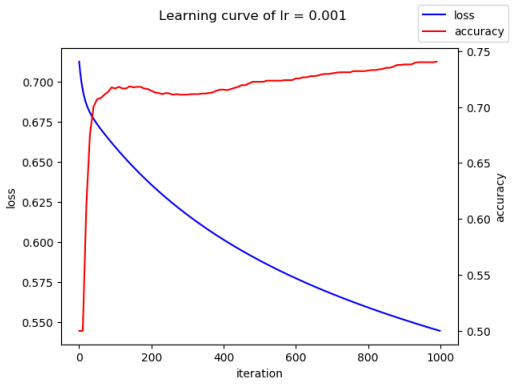
(f) Based on our plots (Figures 3, 4), the logistic regression model learns something about the shoulder parts to distinguish the coat and something about the sleeves to distinguish the pullover.

(g) Inference step of this learning approach is more efficient than the nearest neighbors search which has to find the k nearest neighbors. Another advantage of this learning approach is that it returns the probability of how confident it is that the value belongs to a certain class. The Nearest Neighbor algorithm is non-parametric, hence it is prone to high dimensional data(due to "single" param L2-distance messure), in contrast learning approaches are able to handle more complex(high dim) problems.

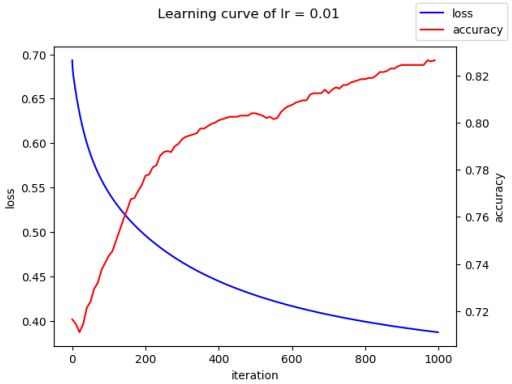
Driton Goxhufi 4233242
Damir Ravlija 5503184



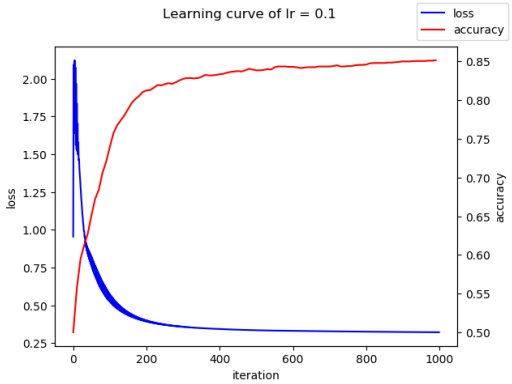
(a) Learning rate set to 0.0001



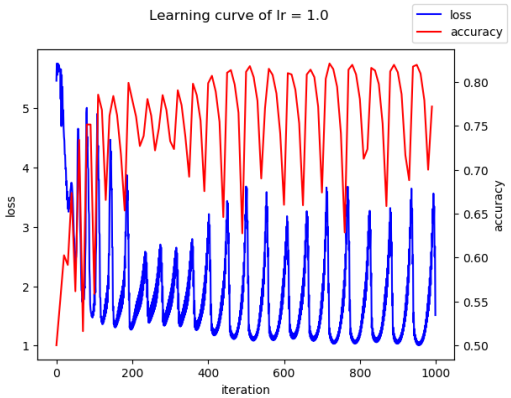
(b) Learning rate set to 0.001



(c) Learning rate set to 0.01



(d) Learning rate set to 0.1



(e) Learning rate set to 1

Figure 2: Plot of results from task (e)

Driton Goxhufi 4233242
Damir Ravlija 5503184

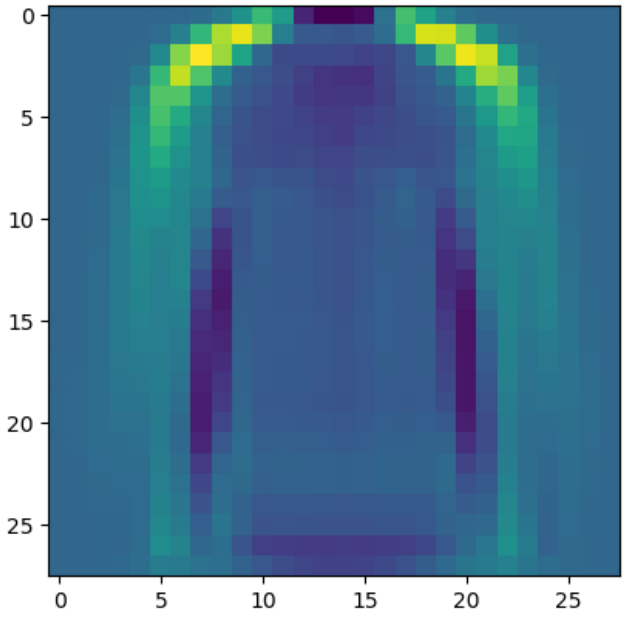


Figure 3: Subtask (f), plot of the optimized weights $\boldsymbol{w}$ as a 28×28 image

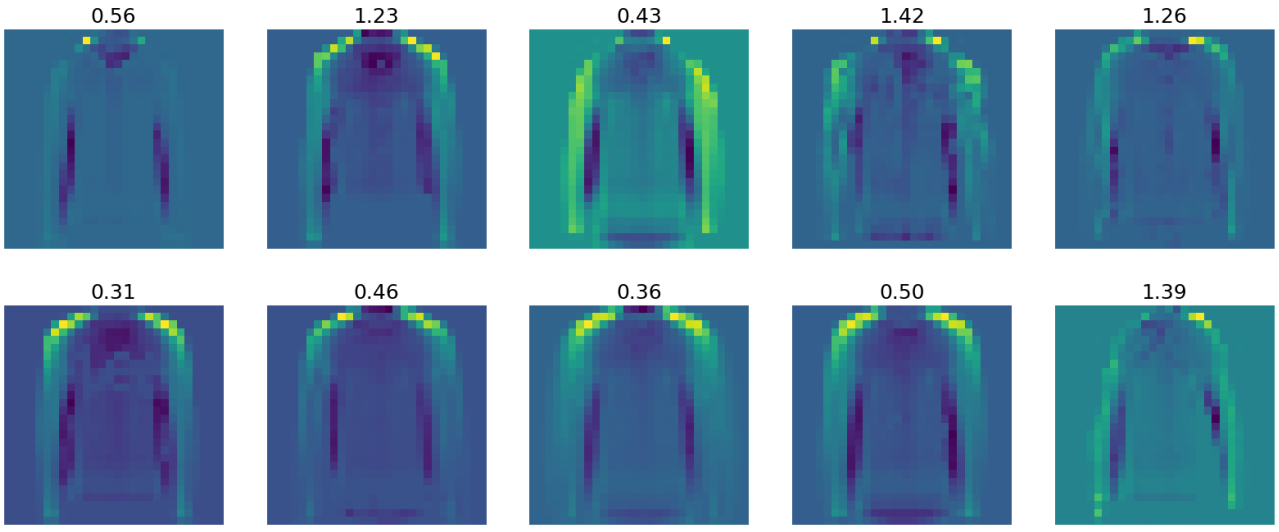


Figure 4: Subtask (f), plot of $\boldsymbol{w} \odot \boldsymbol{x}$. Pullover images in the upper row, coat images in the lower row.