Some questions

1. what is the function of **regularization penalty** $R(W) = \lambda \sum_k \sum_j W_{k,j}^2$?

Improve the generalization performance of the classifiers on test data and lead to less *overfitting*. But why? Since our goal is to minimize loss function, W tend to be diffuse which means no input dimension can have a very large influence on the scores by is itself. For example, consider input vector $x = [1, 1, 1, 1]$ and two weight vectors $w_1 = [1, 0, 0, 0]$ $w_2 = [0.25, 0.25, 0.25, 0.25]$. We will get the same dot product. With **regularization penalty**, we tend to get w_2, since it will lead to less loss score. But why this is a better choice? input vector x has four dimensions, every dimension contains some information. Weights can help these information flow in neural networks. If weights are set to zero, we tend to lose information from corresponding dimensions. Also we want our model to generalize as many situations as possible. Thus we shouldn't overemphasize information from a certain dimension.

2. What is (Multiclass Support Vector Machine)SVM loss, what is it trying to do?

For input vector x_i whose label is y_i. The score $s = f(x_i, W)$. Suppose the score for the j-th class is s_j. Our goal is to make loss as small as possible, so we would prefer each item to become zero which means $s_j + \Delta \leq s_{y_i}$. Right now we can see that SVM "wants" the correct class to have a score higher than incorrect classes by Δ. If we think a higher score means the input date's label belonging to the corresponding class, it makes sense. **But I don't know why this works in theory**.

$$L = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + \Delta)$$

$max(0, -)$ is often called the **hinge loss**.

3. What do parameters Δ and λ control?

- when Δ is small the difference between the correct class score and incorrect class score tend to be low.
- λ will influence the value of W, if W is small, we will get relatively small score, so the margin between scores will be lower.
- All in all, the two parameters control the tradeoff between the data loss and the regularization loss in the objective.

4. What is Softmax classifier and its probabilistic interpretation?

We keep the mapping function $f(x_i; W) = Wx_i$ unchanged but we now interpret these scores as unnormalized log probabilities(**Why we can do this?**) for each class and replace the *hinge loss* with a **cross-entropy loss**.

$$L_i = -log(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}})$$

$f_j(z) = \frac{e^{f_{z_j}}}{\sum_k e^{z_k}}$ is called softmax function: It takes a vector and squashed to a vector of values between zero and one that sum up to one. We can just interpret each value in score vector as the probability of the input vector belongs to its corresponding class. Our goal is to maximize the correct probability, we can achieve by minimizing the negative log likelihood of the correct class. This is just an intuition. **I still don't know the full details of this derivation.**