1. (a) False. The testing error is also a random variable that may or may not be smaller than the training error.

   (b) False. The bias will be increased.

   (c) True/False. Depending on the scenario, $\mathbb{L}_1$ regulerization promotes sparsity, where $\mathbb{L}_2$ promotes convexity.

2. (a) **Non-negativity**

   Non-negativity is satisfied from the definition.

   Zero identity

   $\sqrt{|x-y|} = 0$ if and only if $|x-y| = 0$.

   Symmetry

   From the property of absolute value.

   Triangular inequality

   $(d(x,z)+d(y,z))^2 \geq d(x,z)^2 + d(y,z)^2 = |x-z| + |y-z| \geq |x-z+z-y| = d(x,y)^2$.

   (b) No. For $x = 2, y = 0, z = 1$, $d(x,y) = 4 \geq d(x,z) + d(y,z) = 2$.

   (c) No. KL divergence is not symmetric.

3. (a) The likelihood of the training set is the product of the probabilities of the $y^{(i)}$s given the $x^{(i)}$s:

$$L_\theta(y|x) = \prod_{i=1}^{m} p(y^{(i)}|\theta^T x^{(i)}, 1) = \prod_{i=1}^{m} (\frac{1}{2}\exp(-|\theta^T x^{(i)} - y^{(i)}|)) \tag{1}$$

   (b) Let X be the $m \times p$ data matrix where $X_i = x^{(i)}$, and p is the dimension of the data. The loss function is the negative log-likelihood plus a penalty term, with constant removed,

$$\mathcal{L}(\theta) = \mathbb{1}^\intercal(|X\theta - y|) + ||\theta||_1, \tag{2}$$

   where $\mathbb{1}$ is a $m \times 1$ vector whose entries are all 1.

   (c) By chain rule

$$\nabla\mathcal{L}(\theta) = X^\intercal(\mathbf{sgn}(X\theta - y) \odot \mathbb{1}) + \mathbf{sgn}(\theta), \tag{3}$$

   where **sgn** is the sign function, and $\odot$ is elementwise multiplication.

   The update rule is hence

$$\theta^{(k+1)} = \theta^{(k)} - \eta^{(k+1)}[X^\intercal(\mathbf{sgn}(X\theta - y) \odot \mathbb{1}) + \mathbf{sgn}(\theta)], \tag{4}$$

   where $\eta^{(k+1)}$ is the step length at $k+1$ step.

4. (a) Beta distribution is a conjugate prior for binomial distribution.

(b) The condition distribution

$$p(y_B|\theta_B) = \theta_B^{y_B}(1-\theta_B)^{n_B-y_B}, \tag{5}$$

and the posterior is

$$p(\theta_B) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta_B^{\alpha-1}(1-\theta_B)^{\beta-1}. \tag{6}$$

(c) The posterior given prior parameters α, β is

$$Beta(\alpha + y_B, \beta + n_B - y_B). \tag{7}$$

(d) The log likelihood of a is data set $\log L(\theta_B) = (n_B - y_B)\log(1-\theta_B) + y_B \log\theta_B$. Then $\frac{d\log L(\theta_B)}{d\theta_B} = \frac{n_B-y_B}{\theta_B-1} + \frac{y_B}{\theta_B} = 0$, when $\theta_B = y_B/n_B$. And the second derivative at that point is $\frac{n_B^3}{y_B(y_B-n_B)} < 0$, if $y_B \neq 0$ *or* n_B. And when $y_B = 0$ *or* n_B, graphically the likelihood function decreases or increases monotomically. Hence the MLE estimator when $y_B = 0, 1, ..., n_B$ is always y_B/n_B. In this case, the estimated parameter is $\hat{\theta}_B = 0/40 = 0$. Hence under such condition, y_B/n_B is a scaled binomial distribution with variance $n_B(1-\hat{\theta}_B)\hat{\theta}_B/n_B^2 = 0$, suggesting that the confidence interval with respect to all possible significance level is always $[0, 0]$, which does not reflect the real uncertainty of the estimation at all.

5. Example regularization techniques include but not limit to

   (a) Dropout. Drop out can also be used heuristically for uncertainty.

   (b) Batch normalization. Batch normalization reduces internal covariate shift.

   (c) Early stop. Early stop prevents data overfitting.

   (d) Weight decay. Weight decay reduces variance.

6. (a) The joint entropy of multiple categorical random variables is the same as the entropy of a single categorical random variable with the same set of probabilities. So, in this case, the entropy is $-(0.4\log(0.4) + 0.3\log(0.3) + 0.2\log(0.2) + 0.1\log(0.1)) \approx 1.27$.

   (b) The conditional entropy

$$\mathcal{H}(Y|X) = \sum p(x)H(Y|X=x) \tag{8}$$

$$= 0.7(-4/7\log(4/7) - 3/7\log(3/7)) + 0.3(-2/3\log(2/3) - 1/3\log(1/3)) \tag{9}$$

$$\approx 0.669. \tag{10}$$

7. (a) A convolutional layer with N input channels, M output channels, and $K \times K$ spatial extent requires MNK^2 weights. Hence, we need: $10 \times 20 \times 3 \times 3 = 1800$ weights.

   (b) The locally connected layer has the same pattern of connections as the convolution layer but each of the $5 \times 5 = 25$ output locations will have its own separate set of weights. Hence, the total number of weights is $25 \times 1800 = 45000$.

8. (a) The KL divergence

$$KL(p||q) = \int_{\mathbb{R}^n} \log p dp - \log q dp \tag{11}$$
$$= -\mathcal{H}(p) + \frac{1}{2}\int_{\mathbb{R}^n} (x-\mu)^\intercal (x-\mu) dp + C \tag{12}$$
$$= -\mathcal{H}(p) + \frac{1}{2}\mathbb{E}_p X^\intercal X - \mu^\intercal E_p(X) + \frac{1}{2}\mu^\intercal \mu, \tag{13}$$

where $dp = p(d)dx$, C is the natural log of normalization constant.

(b) The minimizer of (13) $\mu^\star = \mathbb{E}_p(X)$

9. (a) For convenience we assume $p = 1$. Then the output is $\begin{bmatrix} 1 & 4 & 9 \\ 4 & 9 & 16 \end{bmatrix}$.

(b) Let the shape of params be $(a, ...)$ suppose it is of high dimension. Then the ourput shape should be $(m, n, a, ...)$. The two in_axes arguments vectorize along the column and then the row. If params is a scalar, then the output shape is (m, n).

(c) Given input x^1, x^2, the output

$$y_{i,j} = f(x^1_j, x^2_i, \text{params}). \tag{14}$$

The function can be used for mesh-like elementwise operation.

(d) Given input x^1, x^2, the output

$$y_{i,j} = f(x^1_i, x^2_j, \text{params}). \tag{15}$$

The output is different if the function f is not symmetric, or the input shapes of xs1, xs2 do not match, or both.

(e) The code should look like

```
def multi_map(f, xs, xs2, xs3, params):
    return vmap(vmap(vmap(f,in_axes=[0,None,None,None]),in_axes=[None,0,None,None]),
                in_axes=[None,None,0,None])(xs,xs2,xs3,params)
```