Assignment #3: Small Data

In this assignment, we'll look at two approaches to dealing with having small amounts of data. You can use automatic differentiation in your code.

Data preparation Binarize the MNIST dataset. In this assignment, we'll use only **300 examples** in our training set. We'll keep the test set the same size, at 10000 examples.

Problem 1 (L2-Regularized Logistic Regression, 10 points)

In this question, we'll attempt to regularize logistic regression to deal with having such a small dataset. Recall that the likelihood given by this model is:

$$p(c|\mathbf{x}, \mathbf{w}) = \frac{\exp(\mathbf{w}_c^T \mathbf{x})}{\sum_{c'=0}^{9} \exp(\mathbf{w}_{c'}^T \mathbf{x})} \tag{1}$$

(a) Using your code from assignment 2, fit a maximum likelihood estimate of logistic regression to the 300 training points, and report the training and test-set error. Also plot the learned parameters as a set of 10 images.

(b) Next, let's define a prior distribution on parameters, so that we can fit a *maximum a posteriori* (MAP) estimate. Let's consider a spherical Gaussian prior on the parameters:

$$p(\mathbf{w}|\sigma^2) = \prod_{c=0}^{9} \prod_{c=0}^{784} \mathcal{N}(w_{cd}|0, \sigma^2) \tag{2}$$

Write down $\log\left(p(\mathbf{t}|\mathbf{X}, \mathbf{w})p(\mathbf{w}|\sigma^2)\right)$, the log-likelihood of the entire training set $(\mathbf{X}, \mathbf{t})$ of 300 examples, multiplied by the prior on parameters. Also write down its gradient. You do not need to show the derivation. Hint: It should look like the gradient of the training log-likelihood from assignment 2, but with an extra term added that only depends on $\mathbf{w}$.

(c) Fit a MAP estimate of the parameters $\mathbf{w}$ on the training set using gradient ascent. Try different values of σ^2 across several orders of magnitude. For the value of σ^2 with the highest test-set log-likelihood, plot the optimized $\mathbf{w}_{MAP}$ as 10 images. Also print the training and test accuracy, and average predictive log-likelihood:

$$\frac{1}{N} \sum_{i=1}^{N} \log p(t_i|\mathbf{x}_i, \mathbf{w}) \tag{3}$$

Problem 2 (Bayesian Logistic Regression using Stochastic Variational Inference, 20 points)
In this question, we'll avoid choosing a single set of parameters $\hat{\mathbf{w}}$. Instead, we'll approximately *integrate over all possible* $\mathbf{w}$. This will avoid over-fitting by making approximately Bayes-optimal predictions, given the assumptions of our model. The Bayes-optimal predictions are given by:

$$p(c|\mathbf{x}) = \int p(c|\mathbf{x}, \mathbf{w}) p(\mathbf{w}|\mathbf{t}, \mathbf{X}) d\mathbf{w} \tag{4}$$

The posterior over weights is given by:

$$p(\mathbf{w}|\mathbf{t}, \mathbf{X}) = \frac{p(\mathbf{t}|\mathbf{X}, \mathbf{w}) p(\mathbf{w}|\sigma^2)}{\int p(\mathbf{t}|\mathbf{X}, \mathbf{w}) p(\mathbf{w}|\sigma^2) d\mathbf{w}} \propto p(\mathbf{t}|\mathbf{X}, \mathbf{w}) p(\mathbf{w}|\sigma^2) \tag{5}$$

which is the same quantity whose gradients you derived in question 1. If we could sample from the posterior $p(\mathbf{w}|\mathbf{t}, \mathbf{X})$, we could approximate the Bayes-optimal predictions using simple Monte Carlo:

$$p(c|\mathbf{x}_i) = \int p(c|\mathbf{x}_i, \mathbf{w}) p(\mathbf{w}|\mathbf{t}, \mathbf{X}) d\mathbf{w} \approxeq \frac{1}{S} \sum_{j=1}^{S} p(c|\mathbf{x}_i, \mathbf{w}^{(j)}), \qquad \text{each } \mathbf{w}^{(j)} \sim p(\mathbf{w}|\mathbf{t}, \mathbf{X}) \tag{6}$$

In this question, we'll use stochastic variational inference to approximately sample from $p(\mathbf{w}|\mathbf{t}, \mathbf{X})$. To do this, we'll fit the parameters of an approximate posterior $q(\mathbf{w}|\boldsymbol{\phi})$ to make it as close as possible to the true posterior $p(\mathbf{w}|\mathbf{t}, \mathbf{X})$. We'll use stochastic gradient ascent to fit the variational parameters $\boldsymbol{\phi}$.

(a) Using a fully-factorized Gaussian as the variational posterior, the variational parameters $\boldsymbol{\phi} = (\boldsymbol{\mu}, \boldsymbol{\sigma})$ specify the mean and diagonal variance of the distribution on the weights $\mathbf{w}$:

$$q(\mathbf{w}|\boldsymbol{\phi}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}, \boldsymbol{\sigma}^2 I) = \prod_{c=0}^{9} \prod_{c=0}^{784} \mathcal{N}(w_{cd}|\mu_{cd}, \sigma^2_{cd}) \tag{7}$$

How many parameters $\mathbf{w}$ does this model have? How many variational parameters $\boldsymbol{\phi}$?

(b) Code up SVI for this model. That is, use stochastic gradient ascent to find locally optimal variational parameters maximizing the evidence lower bound:

$$\boldsymbol{\phi}^* = \text{argmax}_{\boldsymbol{\phi}} \, \mathbb{E}_{q(\mathbf{w}|\boldsymbol{\phi})} \left[\log p(\mathbf{t}|\mathbf{X}, \mathbf{w}) p(\mathbf{w}|\sigma^2) - \log q(\mathbf{w}|\boldsymbol{\phi}) \right] \tag{8}$$

using simple Monte Carlo to estimate the expectation.

Following the provided starter code, you need only correctly implement the ELBO estimate and log-probability of parameters given data.

As a sanity check, if you optimize $\mathbb{E}_{q(\mathbf{w}|\boldsymbol{\phi})} \left[\log p(\mathbf{t}|\mathbf{X}, \mathbf{w}) p(\mathbf{w}|\sigma^2) \right]$, your variational mean parameters $\boldsymbol{\mu}$ should converge to your MAP estimate of $\mathbf{w}$ if you use the same σ^2.

(c) Use your code to find $\boldsymbol{\phi}^*$. Compute the average predictive accuracy on the test set using simple Monte Carlo using your approximate posterior and 100 samples (S=100):

$$p(t_i|\mathbf{x}_i) = \int p(t_i|\mathbf{x}_i, \mathbf{w}) p(\mathbf{w}|\mathbf{t}, \mathbf{X}) d\mathbf{w} \approxeq \frac{1}{S} \sum_{j=1}^{S} p(t_i|\mathbf{x}_i, \mathbf{w}^{(j)}), \qquad \text{each } \mathbf{w}^{(j)} \sim q(\mathbf{w}|\boldsymbol{\phi}^*) \tag{9}$$

Play with the prior variance σ^2 to see if you can get a higher test-set accuracy than MAP inference.

(d) Plot, using 10 images for each,

i) The variational posterior means $\boldsymbol{\mu}^*$

ii) The variational posterior standard deviations $\boldsymbol{\sigma}^*$

iii) A single sample from the variational posterior $q(\mathbf{w}|\boldsymbol{\phi}^*)$

Briefly describe what these plots are showing and if they are what you expected.

(e) The above plot for a single sample from $q(\mathbf{w}|\boldsymbol{\phi}^*)$ will be extremely noisy. Consider how our model treats pixels which it never sees 'on' across all training examples. In particular, starting from $\log p(t|w,x)$ show that if $x_d \in B$, the set of pixels which are always off, then the training labels do not effect the optimal variational parameters for those pixels.

(f) In the starter code training loop there commented call to **plot_posterior_contours**. This plots the 2D isocontours of the true posterior (blue) and variational posterior (red) for single dimension of K (contourK) and for weights corresponding to two pixels (px1, px2). Uncomment this function and comment on the following:

i) Does the true posterior change during training? Does the variational posterior?

ii) How does the standard deviation of the prior affect the true posterior (try $\sigma = [1.0, 10., 100.]$)

iii) The default px2 is on the image boundary and unlikely to be on across all training data, how is this demonstrated in the true posterior?

iv) Choose another pixel for px2 that is likely to be on in the training data, how does this change the true posterior surface?

v) Our Gaussian approximate posterior is unimodal. If we were to use an improper flat prior $p(w) = c$, could the true posterior in this model ever have more than one mode?