Bryant Har
netid: bjh254

Q1

Q1a

$$\boxed{\text{F - Logistic Regression}}$$

By inspection, we observe a L2 regularization term and a log-loss function characteristic of logistic regression.

Q1b - i

$$\boxed{\ell \text{ is convex}}$$

We find the second derivative.

$$l = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i \cdot x_i^T w}\right) + \frac{\lambda}{2}||w||^2$$

By chain rule and quotient rule, we arrive at

$$\frac{\partial l}{\partial w} = \frac{1}{n}\sum_{i=1}^{n} -\frac{y_i x_i}{1 + e^{y_i \cdot x_i^T w}} + \lambda w$$

$$\frac{\partial^2 l}{\partial w^2} = \frac{1}{n}\sum_{i=1}^{n}(-y_i x_i)\frac{0 - y_i x_i}{\left(1 + e^{y_i \cdot x_i^T w}\right)^2} e^{y_i \cdot x_i^T w} + \lambda = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i x_i}{1 + e^{y_i \cdot x_i^T w}}\right)^2 e^{y_i \cdot x_i^T w} + \lambda$$

Observe that λ, $e^{y_i \cdot x_i^T w}$, and the squared coefficient term are all always positive for real w or by definition. Therefore, $\forall_w \frac{\partial^2 l}{\partial w^2} > 0$, and l is convex. $\mu = \lambda$.

Q1b - ii

$$\boxed{\text{Yes. } \ell \text{ is } \lambda\text{-strongly convex. } \mu = \lambda}$$

Call the generalized input w', such that we have $l(w')$ where $w' = w + \alpha u$. By chain rule,

$$\frac{\partial l}{\partial \alpha} = \frac{\partial l}{\partial w'}\frac{\partial w'}{\partial \alpha} \implies \frac{\partial^2 l}{\partial \alpha^2} = \frac{\partial^2 l}{\partial w'^2}\frac{\partial w'}{\partial \alpha} + \frac{\partial l}{\partial w'}\frac{\partial^2 w'}{\partial \alpha^2} = \frac{\partial^2 l}{\partial w'^2}\frac{\partial w'}{\partial \alpha} + 0$$

$$\frac{\partial^2 l}{\partial w'^2}\frac{\partial w'}{\partial \alpha} = \frac{\partial^2 l}{\partial w'^2}\frac{\partial}{\partial \alpha}(w + \alpha u)$$

This suggests the derivative depends only on $\frac{\partial^2 l}{\partial w'^2}$, as the latter derivative resolves to $|u|$, a unit vector. Examining $\frac{\partial^2 l}{\partial w'^2}$, we see that the exponential term may evaluate to zero in worst case, meaning

$$\frac{\partial^2 l(w+\alpha u)}{\partial \alpha^2} \geq \frac{1}{n}\sum_{i=1}^{n}(0+\lambda) = \lambda \implies \boxed{\frac{\partial^2}{\partial \alpha^2}\ell(w+\alpha u) \geq \lambda}$$

The loss is λ-strongly convex, where λ the regularization hyperparameter.

Q1b - iii

$$\boxed{\text{Yes; Unique global minimum solution.}}$$

Convex functions have no more than one minimum, and if it exists, it is the global minimum. This minimum is the singular solution. We require $\lambda > 0$ by assumption, so logistic regression is strongly convex. Further, there clearly is at least one solution to the logistic regression problem with L2 regularization. Therefore, there exists a unique minimum solution.

Q1b - iv

$$\boxed{\text{It is lipschitz continuous, but there does not exist a data independent value for the Lipschitz constant.}}$$

$$||\nabla f(u) - \nabla f(v)|| \leq L||u-v||$$

Equivalently

$$||\nabla^2 f|| \leq L$$

$$l(w) = \frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-y_i \cdot x_i^T w)) + \frac{\lambda}{2}||w||^2$$

We previously found the second derivative (Hessian matrix),

$$\nabla^2 l = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i x_i}{1+e^{y_i \cdot x_i^T w}}\right)^2 e^{y_i \cdot x_i^T w} + \lambda$$

Therefore,

$$||\nabla^2 l|| \leq L$$

$$\left|\left|\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i x_i}{1+e^{y_i \cdot x_i^T w}}\right)^2 e^{y_i \cdot x_i^T w} + \lambda\right|\right| \leq L$$

By triangle inequality,

$$\frac{1}{n}\sum_{i=1}^{n}\left|\left|\left(\frac{y_i x_i}{1+e^{y_i \cdot x_i^T w}}\right)^2 e^{y_i \cdot x_i^T w}\right|\right| + ||\lambda|| \leq L$$

Note that, taking the extreme case,

$$\left|\left|\frac{e^{y_i \cdot x_i^T w}}{(1+e^{y_i \cdot x_i^T w})^2}\right|\right| \leq \frac{1}{(1+1)^2} = \frac{1}{4}$$

$$\implies \frac{1}{n}\sum_{i=1}^{n}\frac{1}{4}||y_ix_i||^2 + ||\lambda|| \leq L$$

Recall that $y_i \in \{-1, 1\}$. So $||y_i||^2 = 1$. Then,

$$\frac{1}{4n}\sum_{i=1}^{n}||x_i||^2 + \lambda \leq L$$

$$L = \frac{1}{4n}\sum_{i=1}^{n}||x_i||^2 + \lambda = \boxed{L = \frac{||X||^2}{4n} + \lambda}$$

It is L-smooth for such a value.

Q1c

$$w_{t+1} = w_t - \alpha\nabla l(w_t)$$

$$\boxed{w_{t+1} = w_t + \frac{\alpha}{n}\left(\sum_{i=1}^{n}\frac{y_ix_i}{1+e^{y_i\cdot x_i^T w_t}} + \lambda w_t\right)}$$

Q1d

For a single (batch size of 1) sample j uniformly sampled from $\{1, \dots, n\}$,

$$w = w - \alpha\nabla l(w, x_j, y_j)$$

$$\boxed{w_{t+1} = w_t + \frac{\alpha}{n}\left(\frac{y_jx_j}{1+e^{y_j\cdot x_j^T w_t}} + \lambda w_t\right)}$$

Q1e

$$\boxed{\text{It depends}}$$

In lecture, we proved that for gradient descent to converge, $1 \geq \alpha L$ must be satisfied. Substituting our values for L and $\alpha = \frac{1}{4\lambda}$,

$$1 \geq \left(\frac{||X||^2}{4n} + \lambda\right)\frac{1}{4\lambda}$$

$$1 \geq \left(\frac{||X||^2}{4n} + \lambda\right)\frac{1}{4\lambda}$$

Clearly, the truth of this depends on how large $||X||^2$ is. For sufficiently large $||X||^2$, this condition fails and GD fails to converge. It depends on the inputs.

$$\boxed{\text{Whether it converges depends}}$$

Q1f

$$\boxed{\text{No}}$$

Stochastic gradient descent, especially for batch sizes of 1, requires finer step sizes than normal gradient descent to converge over expectation. It won't necessarily converge for the same step size. If our step size is too large, one bad batch can send our weights outside of the noise ball. From lecture, a step size smaller than what is required for GD will guarantee convergence over expectation to some sufficiently small noise ball.

Q2

Throughout, let $\mathbb{A}$ be the calculation accumulator, simply indicating that a portion of the work has been computed.

2a

$$h_w(x) = \text{sign}(x^T w)$$

Both $x, w \in \mathbb{R}^d$. We can compute the dot product as multiplying element wise and then summing. This requires d multiplications and $d-1$ pairwise sums, arriving at

$$h_w(x) = \text{sign}(\mathbb{A})$$

$$\text{Op Count: } d + (d-1) = 2d - 1$$

Taking the sign of the resultant scalar takes 1 operation,

$$h_w(x) = \mathbb{A}$$

$$\text{Op Count: } 2d - 1 + 1 = 2d$$

The calculation is complete with $\boxed{\text{num operations} = 2d}$

2b

$$l(w) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i \cdot x_i^T w)) + \frac{\lambda}{2}||w||^2$$

We can consider the $||\cdot||^2$ as a dot product $w \cdot w$. Since $w \in \mathbb{R}^d$, by above, this requires $2d-1$ operations. We then scale the resultant scalar by λ and then by $\frac{1}{2}$. This involves a further 1 multiplication and 1 division.

$$l(w) = \frac{1}{n}\sum_{i=1}^{n} \log(1 + \exp(-y_i \cdot x_i^T w)) + \mathbb{A}$$

$$\text{Op Count: } 2d - 1 + 1 + 1 = 2d + 1$$

From above, $-y_i x_i^T w$ consists first of a dot product of d-dimensional vectors (again, $2d-1$ multiplications). Then, the resultant scalar is scaled by y_i and by -1 (2). Then, for a scalar, exponentiation, incrementing, and taking the log all involve one operation (3),

$$l(w) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{A} + \mathbb{A}$$

$$\text{Op Count:}(2d+1)+(2d-1)+(1+1)+(1+1+1)=4d+5$$

We compute each term n times, then sum all n terms for $n-1$ pairwise sums. Per above, computing each term costs $2d+4$ operations. We then divide the resultant scalar by n, invoking another operation.

$$l(w)=\mathbb{A}+\mathbb{A}$$

$$\text{Op Count: }2d+1+(2d+4)n+(n-1)+1=2d+1+(2d+5)n$$

Adding the two remaining scalar terms,

$$l(w)=\mathbb{A}$$

$$\text{Op Count: }2d+1+(2d+5)n+1=(2d+5)n+2d+2$$

The computation is complete with $\text{num operations}=\boxed{(2d+5)n+2d+2}$

2c

Having cleared up most computation types, we omit most of the analysis. From above,

$$w_{t+1}=w_t+\frac{\alpha}{n}\left(\sum_{i=1}^{n}\frac{y_ix_i}{1+e^{y_i\cdot x_i^Tw_t}}+\lambda w_t\right)$$

We begin.
Operations: 0

$$w_{t+1}=w_t+\frac{\alpha}{n}\left(\sum_{i=1}^{n}\mathbb{A}x_i+\lambda w_t\right)$$

We dot product (2d-1), scale (1), exponentiate (1), add (1), divide y_i by the result (1),
Operations: $2d+3$

$$w_{t+1}=w_t+\frac{\alpha}{n}\left(\sum_{i=1}^{n}\mathbb{A}+\lambda w_t\right)$$

Generating the coefficient and scaling each of the $x_i,\dots x_n$ vectors costs $(2d+3+d)n=(3d+3)n$.
Operations: $(3d+3)n$

$$w_{t+1}=w_t+\frac{\alpha}{n}(\mathbb{A}+\mathbb{A})$$

Adding n vectors costs $(n-1)$ operations for each dimension. Combine the L2 term.
Operations: $(3d+3)n+d(n-1)+d=(4d+3)n$

$$w_{t+1}=w_t+\mathbb{A}$$

Adding vectors (d), then scaling by α (d), and dividing by n (d),
Operations: $(4d+3)n+d+d+d=(4d+3)n+3d$

$$w_{t+1}=\mathbb{A}$$

Adding the final two vectors,
Operations: $(4d+3)n+3d+d=(4d+3)n+4d$
The computation is complete with $\text{num operations} = \boxed{(4d+3)n+4d}$
This could of course be done faster by computing constant coefficients first before scaling vectors.

2d

From above, for batch size 1,

$$w_{t+1} = w_t + \frac{\alpha}{n}\left(\frac{y_j x_j}{1+e^{y_j \cdot x_j^T w_t}} + \lambda w_t\right)$$

We begin.
Operations: 0

$$w_{t+1} = w_t + \frac{\alpha}{n}\left(\mathbb{A} + \mathbb{A}\right)$$

Reusing the above result (2d+3+d) and combining the L2 term (2d),
Operations: $2d+3+d+d=4d+3$

$$w_{t+1} = w_t + \mathbb{A}$$

Combining the inner term and scaling the resulting vector by α and dividing by n,
Operations: $4d+3+d+d+d=7d+3$

$$w_{t+1} = \mathbb{A}$$

Adding the final two vectors,
Operations: $7d+3+d=8d+3$

The computation is complete with $\text{num operations} = \boxed{8d+3}$. We verify that this corresponds to our previous result $(4d+3)n+4d$, when $n=1$.

2e

From 2c, we have

$$\text{num operations} = (4d+3)n+4d$$

Substituting $d=784$, $n=60000$, we require $188,343,136$ computations per gradient descent iteration.

Given $60 \cdot 10^{12}$ flops per second at max performance, one hour is 3600 seconds. This is,

$$60 \cdot 10^{12} \cdot 3600 = 2.16 \cdot 10^{17} \text{ computations}$$

Dividing the two,

$$\frac{2.16 \cdot 10^{17}}{188343136} = 1.15 \cdot 10^{9}$$

It can do about $\boxed{10^9 \text{ iterations}}$ in an hour