1.1

SVMs find the optimal hyperplane that best separates classes in a dataset; thereby maximizing the margin, distance between the hyperplane and the nearest data point. Logistic regression struggles in such cases as estimating weights become unstable with higher-dimensional data.

We can also use the kernel trick i.e. use kernel function with SVMs. It helps transform data into higher-dimensional space where a linear boundary can separate the classes (which were non-linearly separable).

SVMs tend to avoid overfitting while logistic regression relies on regularization to avoid overfitting. In a high dimensional space. logistic regression may struggle to find a good fit.

1.2

For n training samples, the kernel matrix requires $O(n^2)$.SVMs overcome the memory issue by using Support Vectors. Only a subset of training points determines the final decision boundary. As a result, only the support vectors need to be stored, which reduces memory requirements.

Also, each training sample is associated with a Larange multiplier. For non-support vectors, the multiplier values are zero. As a result, the decision boundary can be expressed only in terms of the support vectors while discarding all the other training points after training.

2.1

i) Input layer: $\mathbf{x} \in \mathbb{R}^d$; d = # of input features.

ii) Hidden layer: $h \in \mathbb{R}^m$; m = # of hidden units.

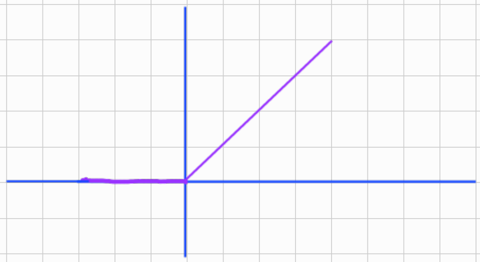
The ReLU function is defined as: $ReLU(x) = \begin{cases} 1 ; & x > 0 \\ 0 ; & x \leq 0 \end{cases}$

$ReLU(z) = \max(0, z)$.

$h = ReLU(W_1 \mathbf{x} + b_1)$.

; $W^{(1)} \in \mathbb{R}^{m \times d}$ is the weight matrix.

$b^{(1)} \in \mathbb{R}^m$ is the bias vector.



iii) Output layer: $\hat{y} \in \mathbb{R}$. which is a scalar value for regression.

$\hat{y} = W^{(2)} h + b$; $W^{(2)} \in \mathbb{R}^m$

$b^{(2)} \in \mathbb{R}^k$

Forward Pass

$$z^{(1)} = W^{(1)} x + b^{(1)}$$

$$h = ReLU(z^{(1)}) = \max(0, z^{(1)})$$

$$z^{(2)} = W^{(2)} h + b^{(2)}$$

$$\hat{y} = z^{(2)}$$

Loss function

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} (y_i - \hat{y})^2$$

; $\hat{y}$ is the predicted value
y is the true target.

For a single sample, $\mathcal{L} = \frac{1}{2} (y - \hat{y})^2$.

Back-propagation

Find gradients of the loss $\mathcal{L}$ w.r.t. all the parameters

$\theta \leftarrow \theta - \alpha \frac{\partial \mathcal{L}}{\partial \theta}$. ; α = learning rate
θ = parameters to be updated.

Gradient Calculation

1. Gradient w.r.t output ($\hat{y}$) : $\frac{\partial d}{\partial \hat{y}} = \hat{y} - y = \delta^{(2)}$

2. Gradient w.r.t $z^{(2)}$: $\frac{\partial d}{\partial z^{(2)}} = \frac{\partial d}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z^{(2)}} = \hat{y} - y = \delta^{(2)}$

3. Gradient w.r.t $W^{(2)}$:

$$z^{(2)} = W^{(2)} h + b^{(2)}$$

$$\frac{\partial z^{(2)}}{\partial W^{(2)}} = h \quad ; \quad \frac{\partial d}{\partial W^{(2)}} = \frac{\partial d}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial W^{(2)}}$$

$$= \delta^{(2)} h^{T}$$

4. Gradient w.r.t $b^{(2)}$: $\frac{\partial z^{(2)}}{\partial b^{(2)}} = 1 \; ; \; \frac{\partial d}{\partial b^{(2)}} = \frac{\partial d}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial b^{(2)}} = \delta^{(2)}$.

5. Gradient w.r.t h : $z^{(2)} = W^{(2)} h + b^{(2)}$.

$$\frac{\partial z^{(2)}}{\partial h} = W^{(2)} \quad ; \quad \frac{\partial d}{\partial h} = \frac{\partial d}{\partial z^{(2)}} \cdot \frac{\partial z^{(2)}}{\partial h}$$

$$= \delta^{(2)} W^{(2)}.$$

6. Gradient w.r.t $z^{(1)}$:

$$h = ReLU(z^{(1)}).$$

$$\frac{\partial h}{\partial z^{(1)}} = \begin{cases} 1 & ; z^{(1)} > 0 \\ 0 & ; z^{(1)} \le 0 \end{cases} \rightarrow a'(z^{(1)})$$

$$\frac{\partial d}{\partial z^{(1)}} = \frac{\partial d}{\partial h} \cdot \frac{\partial h}{\partial z^{(1)}} = W^{(2)^{T}} \delta^{(2)} \odot a'(z^{(1)}) = \delta^{(1)}$$

7. Gradient w.r.t $W^{(1)}$:

$$z^{(1)} = W^{(1)} x + b^{(1)}$$

$$\frac{\partial z^{(1)}}{\partial W^{(1)}} = x \quad ; \quad \frac{\partial d}{\partial W^{(1)}} = \frac{\partial d}{\partial z^{(1)}} \cdot \frac{\partial z^{(1)}}{\partial W^{(1)}}.$$

$$= \delta^{(1)} x^{T}$$

8. Gradient w.r.t $b^{(1)}$:

$$\frac{\partial z^{(1)}}{\partial b^{(1)}} = 1 \; ; \; \frac{\partial d}{\partial b^{(1)}} = \frac{\partial d}{\partial z^{(1)}} \cdot \frac{\partial z^{(1)}}{\partial b^{(1)}} = \delta^{(1)}.$$

2.2

Some possible data augmentation strategies could be:

a. Shifting the pickup and drop off coordinates within a small radius by small increments
b. Shift pickup and drop off time stamps by small increments
c. Perturb passenger count by 1 or 2

NYC's grid layout can bring changes in trip duration with this variation preventing overfitting to exact pickup/drop-off time and coordinates. Trip duration is also affected by the time of day. Shifting time stamps during rush traffic or away from it helps generalize data points.

3.1.

Gradient boosting constructs a prediction model $F(x)$ as an additive combination of weak learners.

$$F(x) = F_0(x) + \eta h_1(x) + \eta h_2(x) + \cdots + \eta h_M(x).$$

; $F_0(x)$: initial model
$h_m(x)$: weak learners added at each iteration m.
η : learning rate of each weak learner.
M : # of iterations

The goal is to minimize a loss function $L(y, F(x))$.

1. $F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma)$

2. Add models:

$$r_{im} = \left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right]_{F(x_i) = F_{m-1}(x_i)}$$

; r_{im} : direction the model needs to improve to reduce loss.

Train weak learner $h_m(x)$ to predict the residuals r_{im}

$$h_m(x) = \arg\min_{h} \sum_{i=1}^{n} (r_{im} - h(x_i))^2$$

. Update models:

$$F_m(x) = F_{m-1}(x) + \eta h_m(x).$$

3. final model

$$F_M(x) = F_0(x) + \eta \sum_{m=1}^{M} h_m(x).$$

Gradient boosting minimizes the loss function by iteratively fitting the weak learners to the -ve gradient of the loss. Each step reduces the error by targeting the residuals and the iterative updates add to the improvement. This results towards a minimum loss.

3.2.

$$l(y, F(x)) = \log(1 + e^{-2y F(x)}) \; ; \qquad (1)$$

$y \in \{-1, 1\}$
$F(x) = \frac{1}{2} \log \frac{1+\hat{y}}{1-\hat{y}}$

The probability of $y \in \{-1,1\}$ given input x is:

$$P(y=1 \mid x) = \frac{1}{1+e^{-F(x)}}$$

$$P(y=-1 \mid x) = 1 - P(y=1 \mid x) = 1 + e^{F(x)}.$$

The p.m.f is given as.

$$P(y \mid x) = P(y=1 \mid x)^{\mathbb{I}(y=1)} \cdot P(y=-1 \mid x)^{\mathbb{I}(y=-1)}$$

Log-likelihood & Loss function

$$\log P(y \mid x) = \mathbb{I}(y=1) \log P(y=1 \mid x) + \mathbb{I}(y=-1) \log P(y=-1 \mid x).$$

$$\log P(y \mid x) = \mathbb{I}(y=1) \log\left(\frac{1}{1+e^{-F(x)}}\right) + \mathbb{I}(y=-1) \log(1+e^{F(x)}).$$

$$= -\mathbb{I}(y=1) \log(1+e^{F(x)}) - \mathbb{I}(y=-1) \log(1+e^{F(x)}).$$

$\because y \in \{-1, 1\}$.

$$\log P(y \mid x) = -\log(1+e^{-y F(x)}).$$

$$\therefore l(y, F(x)) = \log(1+e^{-y F(x)}) \qquad (2)$$

When (2) is scaled by a factor of 2, we get (1).

The logit $F(x)$ is defined as:

$$F(x) = \frac{1}{2} \log\left(\frac{1+\hat{y}}{1-\hat{y}}\right) \; ; \; \hat{y} \text{ is the predicted probability of } y = 1.$$

$$F(x) = \log\left(\frac{P(y=1 \mid x)}{P(y=-1 \mid x)}\right)$$

Substituting values of $P(y=1 \mid x)$ and $P(y=-1 \mid x)$

$$F(x) = \log\left(\frac{\frac{1}{1+e^{-F(x)}}}{\frac{1}{1+e^{F(x)}}}\right) = \log\left(\frac{1+e^{F(x)}}{1+e^{-F(x)}}\right)$$

$$\therefore F(x) = \log(e^{F(x)})$$