6. Linear Models part 1

M.A.Z. Chowdhury and M.A. Oehlschlaeger

Department of Mechanical, Aerospace and Nuclear Engineering
Rensselaer Polytechnic Institute
Troy, New York

chowdm@rpi.edu
oehlsm@rpi.edu

"To use a linear model or not to use a linear model that is the question?"
- Hamlet, Fictional Prince of Denmark

MANE 4962 and 6962

Linear combination

- The heart of linear algebra is in two operations. Both with vectors.
- We add vectors to get $\underline{v} + \underline{w}$.
- We multiply them by scaler numbers c and d to get $c\underline{v}$ and $d\underline{w}$.
- Combining those two operations gives the linear combination $c\underline{v} + d\underline{w}$.
- We commonly use **boldface** or underline to represent vectors.
- $\underline{v}$ and $c\underline{v}$ lie on the same straight line.

$$c\underline{v} + d\underline{w} = c\begin{bmatrix}1\\1\end{bmatrix} + d\begin{bmatrix}2\\3\end{bmatrix} = \begin{bmatrix}c+2d\\c+3d\end{bmatrix}$$

Why study linear models?

- In ML, for many cases, we may not be privy to the implementation details.

But having a good understanding of how linear models work can help you do the following:

- quickly zero in on the appropriate model
- Find right training algorithm
- Find good set of hyperparameters for the task
- Help debug issues
- Perform efficient error analysis

Linear models are very important if you want to understand, build, and train neural networks

Linear Hypothesis

- Consider n independent variables or features
- we want to approximate target function $y = f(\underline{x}) = f(x_1, x_2, \ldots, x_n)$
- We can use weights (model parameters), $\underline{w}$
- $\hat{y} = h(\underline{x}; \underline{w})$ is the hypothesis function to approximate the target function.
- $\hat{y} = h(\underline{x}; \underline{w}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots + w_n x_n$
- $\hat{y} = h(\underline{x}; \underline{w}) = w_0 x_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \ldots + w_n x_n \quad [x_0 = 1]$
- $\hat{y} = h_{\underline{w}}(\underline{x}) = \underline{w} \cdot \underline{x}$ [Another notation]
- If both $\underline{w}$ and $\underline{x}$ are column vectors then,
- $\hat{y} = \underline{w}^T \cdot \underline{x}$
- $y \in \mathbb{R}, \hat{y} \in \mathbb{R}$
- $x \in \mathbb{R}^n$
- Sometimes the underline is omitted and it is understood from context whether the notations are referring to a vector or a vector component.

a linear model makes a prediction by simply computing weighted sum of the input features, plus a constant called the bias or intercept

Linear Hypothesis

Consider, m data points in the form given below, and we want to predict y from a single input variable or feature x

input	target	prediction
x_1	y_1	$\hat{y}_1$
x_2	y_2	$\hat{y}_2$
$\vdots$	$\vdots$	$\vdots$
x_m	y_m	$\hat{y}_m$

- $\hat{y} = h(\underline{x}; \underline{w}) = w_0 + w_1 x_1$ [for $n = 1$, single feature, we have one input variable]
- x_1 is the input variable/feature not the data point in the equation.
- w's are called model parameters (weights), n is the number of feature
- Model parameters and hyperparameters are not same. Model parameters depend on training data, hyperparameters are chosen, or set, or optimized.

Cost function

How to train a linear model?

$\rightarrow$ estimate mistakes and correct them via error measure.

Cost function measures the total amount of incorrect predictions across the data points.

MSE is a good measure of the overall mistakes made by the model and can be used as cost function. We will modify it slightly though for the sake of math. m = total number of data points

- squared error (loss), $\mathrm{L} = (y - \hat{y})^2$
- mean squared error, $mse = \frac{1}{m}\sum_i (y_i - \hat{y}_i)^2$
- cost function, $J = \frac{1}{2m}\sum_i (y_i - \hat{y}_i)^2$

Cost function

$$J = \frac{1}{2m}\sum_i (y_i - \hat{y}_i)^2$$

$$\implies J = \frac{1}{2m}\sum_i (y_i - \hat{y}_i)^2$$

We know, $\hat{y}_i = w_0 + w_1 x_i$ for linear model, so

$$J = \frac{1}{2m}\sum_i \{y_i - (w_0 + w_1 x_i)\}^2$$

Finally, we get

$$J(x; \underline{w}) = \frac{1}{2m}\sum_i \{y_i - (w_0 + w_1 x_i)\}^2$$

$J(x; \underline{w}) = J_w(x)$ [Another notation]

Cost function

$$J(x; \underline{w}) = \frac{1}{2m} \sum_i \{y_i - (w_0 + w_1 x_i)\}^2$$

$J(x; \underline{w})$ tells us the total amount of mistakes made by the model on the data.

All we have to do is to minimize J. This will ensure the model is learning.