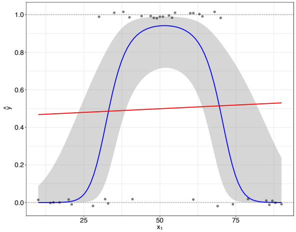
Beyond Linear Decision Boundaries

- Logistic regression naturally yields a sigmoidal relationship between the input features and the response probability.
- In a classification setting this translates to linear (affine) decision boundaries.
- Similarly, multinomial regression used for classification also yields straight line boundaries via convex polytope decision boundaries.
- One may then ask if these shallow neural network models can create other forms of response curves or decision boundaries?

Enhancing the Sigmoidal Response

- Consider a dataset $\mathcal{D}$ where each observation is for a different geographic location and where the feature vector has a single coordinate which is the level of precipitation at that location (in millimeters/month).
- For each observation, the associated $y^{(i)} \in \{0, 1\}$, determines the absence (0) or presence (1) of a certain species.
- At locations i that are not too dry or not too wet, the species tends to be present, whereas when the precipitation is very low or very high the species tends to be absent.



- Could you guess the underlying model ?

$$\phi(x) = \frac{1}{1 + e^{-(b+w_1x+w_2x^2)}}.$$

- Following from basic properties of the parabola $b + w_1x + w_2x^2$, if $w_2 < 0$ then we have that $\phi(x) \to 0$ as $x \to -\infty$ or $x \to \infty$
- $\phi(x)$ is maximized at $x = -w_1/w_2$ which is the maximal point of the parabola.
- Similarly, if $w_2 > 0$ the shape of $\phi(x)$ is reversed and $\phi(x)$ has a minimum point at the minimum point of the parabola. In both cases, $\phi(x)$ is symmetric about $x = -w_1/w_2$.

Polynomial Feature Engineering

- It is quite common to use powers for feature engineering and this makes the linear combination of the engineered features a polynomial.
- When there are initially p input features we can automate the creation of more features by choosing each new feature as a power product or monomial form $x_1^{k_1} x_2^{k_2} \cdot \ldots \cdot x_p^{k_p}$ for some non-negative integers $k_1, k_2, \ldots, k_p$.
- It is common to limit the **degree** by a constant r via

$$k_1 + k_2 + \ldots + k_p \leqslant r.$$

- For example when $r = 2$ the set of engineered features is,

$$\widetilde{x} = (x_1, x_2 \ldots, x_p, x_1^2, x_2^2, \ldots, x_p^2, x_1 x_2, x_1 x_3, \ldots, x_{p-1} x_p).$$

- In this case $\widetilde{x} \in \mathbb{R}^{p(p+3)/2}$ and thus $d = 1 + p(p+3)/2$ for logistic regression.

Polynomial Feature Engineering

- For example if initially $p = 1,000$ features then there are about half a million engineered features with $d = 501,501$.
- For higher degrees of the monomial features $r > 2$ the number of paramater could be astronomic ... (see Chapter 3 of the textbook)
- Solution: we argue that with deeper neural networks one may sometimes get more expressivity without creating such a large number of parameters.