Introduction to Convex Neural Networks

Ahern Nelson, MSU Denver

Introduction

In a regression problem we have values $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$, where x is a vector of variables and y some value related to these variables. In this context our goal is to "explain" the relationship between x and y. We suppose that there exists a function, f, relating x to y with some random error. That is,

$f(x) = y + \varepsilon$, where ε is a random variable

How do we do This?

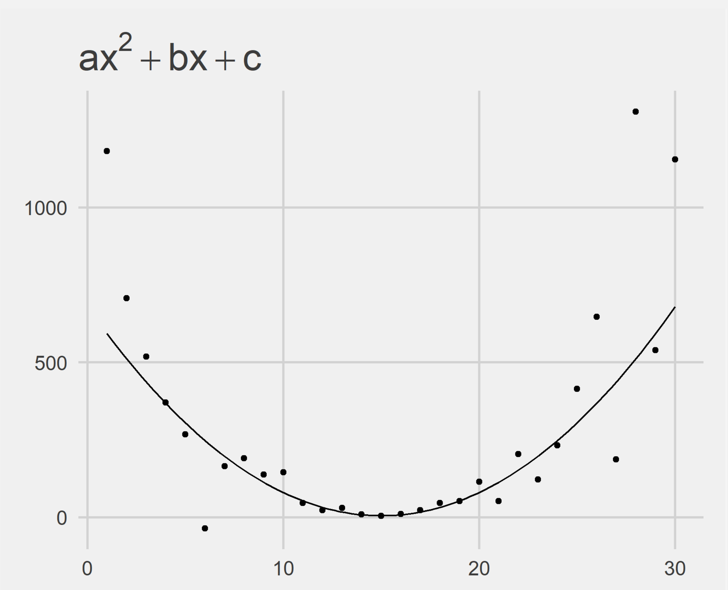
Given a sample of size n we specify a loss function that evaluates how well an estimate of f, $\hat{f}$, fits the data. For example, for an observation (x, y) the squared loss is defined by $Q(\hat{f}(x), y) = (\hat{f}(x) - y)^2$.

We wish to estimate the optimal function, f^*, that minimizes the loss across all observations, i.e.,

$$f^* = \underset{f}{argmin} \sum_{i=1}^{n} (f(x_i) - y_i)^2 \qquad (1)$$

Parametric Regression

Minimizing with respect to any function, f, is vague. Instead, we specify f in a parametric form, e.g., $f(x) = ax^2 + bx + c$ and estimate the unknown parameters, (a, b, c), that form f by minimizing the cost with respect to the parameters.



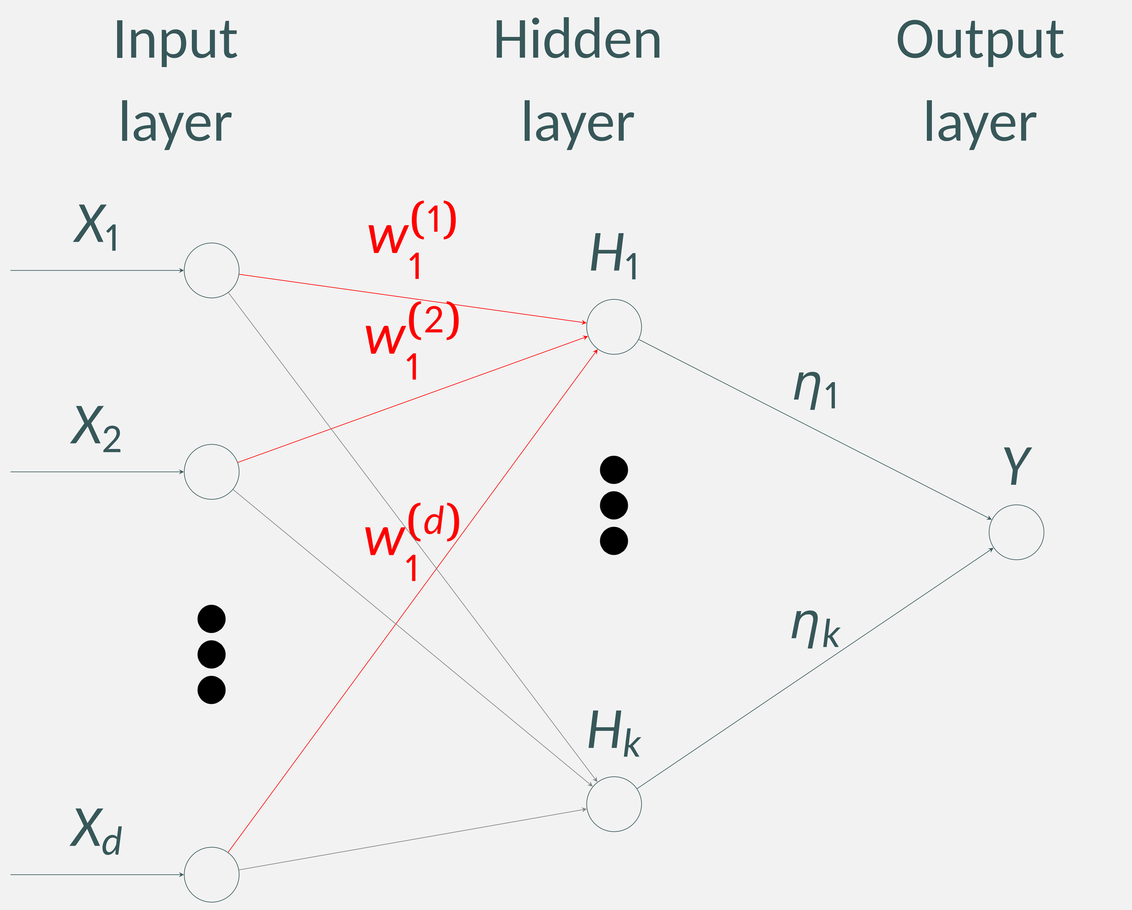
Linear Models

Linear Regression restricts the class models to functions that can be expressed as a linear combination of the unknown parameters. For example, let $x = (1, x_1, \ldots, x_d)$ represent the input variables and let $\beta = (\beta_0, \ldots, \beta_d)$. Then a function of the form $f(x) = \beta^T x = \sum_{i=0}^{d} x_i \beta_i$ can be appropriately estimated by linear methods. However, this condition is potentially restrictive and could lead to severe under fitting in high dimensions.

Single Hidden Layer Networks

Single Hidden Layer Neural Networks alleviate this problem by filtering the parameters through a non-linear transformation. In this problem we specify a nonlinear activation function σ, and an integer k representing the number of hidden units.

When optimizing the cost function on a neural network with respect to the parameters it is unlikely, and often implausible, to find the global minima.

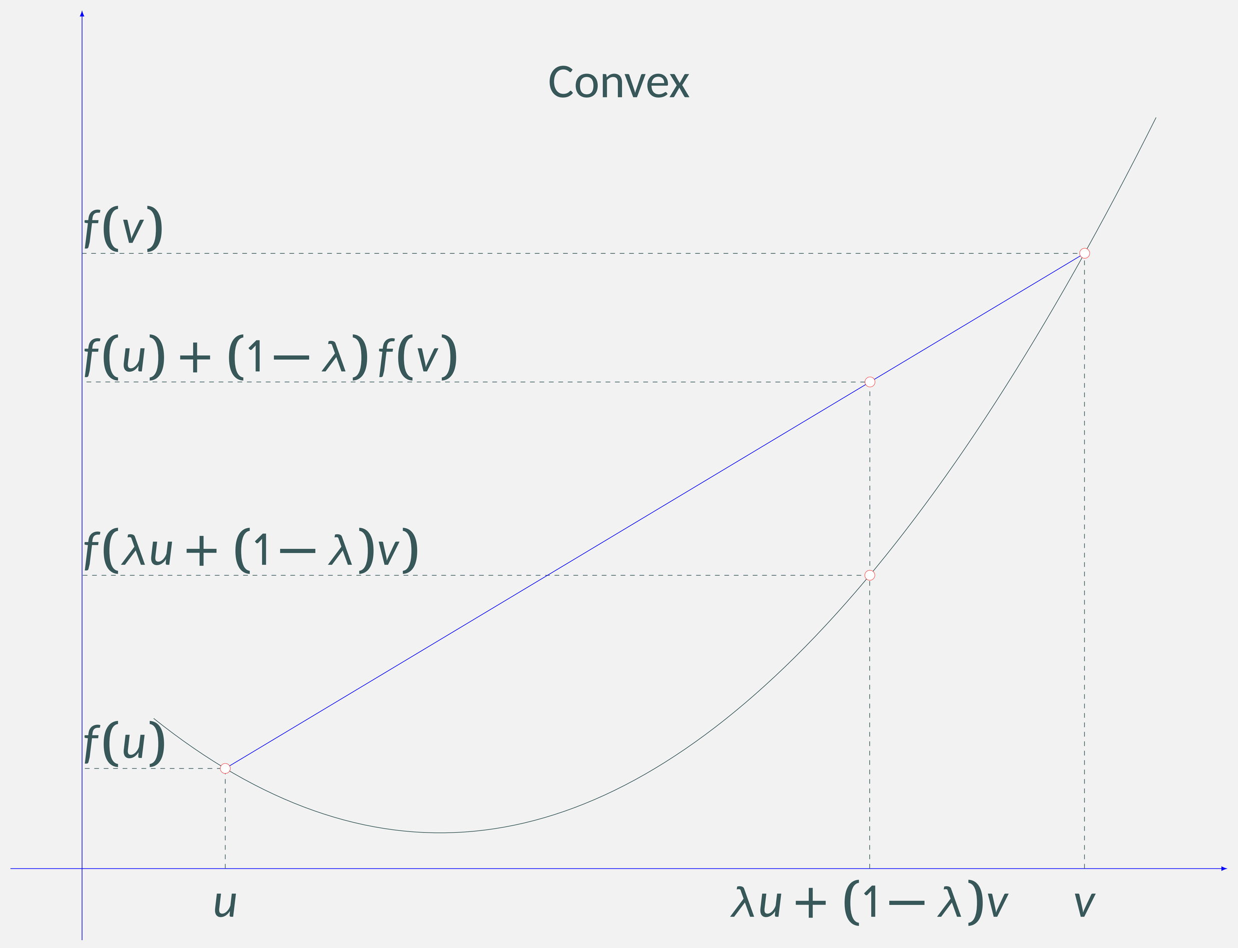


$$f(x) = \sum_{i=1}^{k} \eta_i H_i(x) = \sum_{i=1}^{k} \eta_i \sigma(w_i^T x) \qquad (2)$$

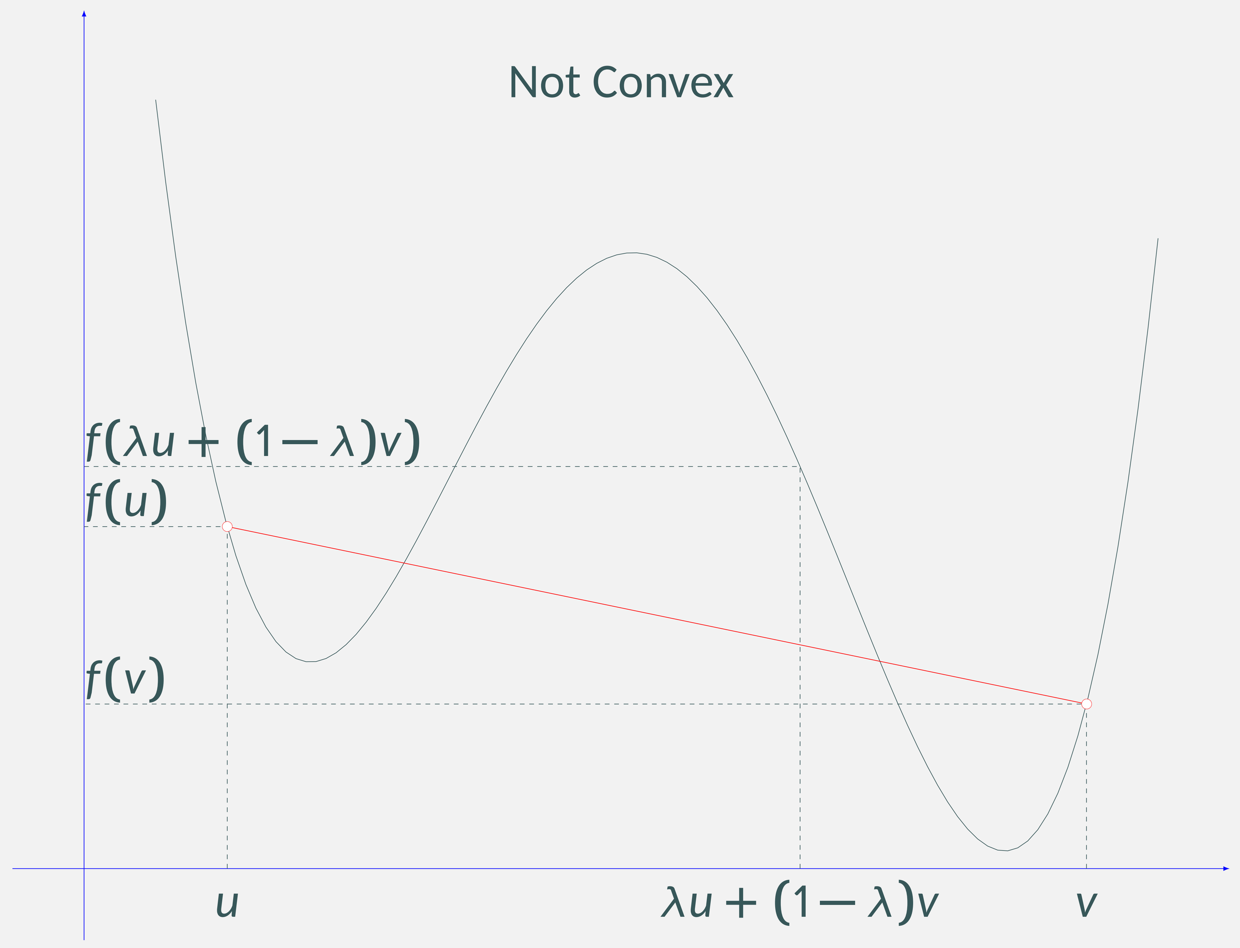
Convex Functions

A function is convex if for any values u, v the following holds:

$$f(\lambda u + (1-\lambda)v) \leq \lambda f(u) + (1-\lambda) f(v), \quad \text{for} \lambda \in [0, 1] \qquad (3)$$



Local Minima = Global Minima



Potential For Many Local Minima

Convex Neural Networks

We can view the parameter estimation as a convex optimization problem by tweaking our perspective. Let $\mathscr{H}$ represent the space of possible hidden unit functions, H_i. Informally this can be thought of as the space of all possible input weights w_i. Consider a network, f, that incorporates all of $\mathscr{H}$.

If $\mathscr{H}$ is known then all we have to estimate is the output weights, η, but the problem is there are as many output weights as there are elements of $\mathscr{H}$ which may be of arbitrary size. However, by using convex regularization tools a finite solution may be obtained.

Regularization

We add a regularization term to the minimization criteria. Regularization penalizes large or complex parameter estimates. In this case we choose a regularizer Ω that is convex in η and promotes sparsity. That is, all but a finite number of η are set to zero. In a sense the number of hidden units is "chosen" in the optimization.

Convexification

For a network, f, as described above, a loss Q convex in the first argument, and a convex regularization term Ω. Then the cost of f for a sample of size n, is convex in the parameter η.

$$C(\mathscr{H}, Q, \Omega, \eta) = \sum_{i=1}^{n} Q(f(x_i), y_i) + \lambda \Omega(\eta) \quad (4)$$

References

[1] P. Vincet O. Dellaleu P. Marcotte Y. Bengio, N. Roux. Convex Neural Networks. *NIPS Proceedings*, 18, 2005.

[2] F. Bach. Breaking the Curse of Dmensionality with Convex Neural Networks.