Idea: Adaption of theory to networks with biases

Leander Kurscheidt

August 1, 2018

Abstract

1 Affine Transformations

Since our NN "somewhat" resembles linear functions following from the rule $\sigma(x) = \sigma'(x)x$ (i see this rule as separating the linear from the non-linear part), i am curious to see whether ideas of affine spaces and how to integrate them into linear-spaces translate.

I think they do.

2 Idea

Affine subspaces are usually define via $A = v + U_V$, where U_V is a subspace of V and v is a vector of V (see (german) wikipedia). But you can work in them using homogenous coordinates.

Every affine transformation can be turned into a linear transformation with a constant 1-input using the Augmented Matrix trick.
For example the intercept in linear models in statistics is often modeled this way.
We can adapt this trick with only one additional constant input using this matrix:

$$W' = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1n} & bias_1 \\ w_{21} & w_{22} & w_{23} & \dots & w_{2n} & bias_2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ w_{d1} & w_{d2} & w_{d3} & \dots & w_{dn} & bias_n \\ 0 & \dots & \dots & \dots & 0 & 1 \end{bmatrix} = \begin{bmatrix} W & & & b \\ 0 & \dots & 0 & 1 \end{bmatrix}$$

the last line is the difference to the usual construction and should be ommitted in the last layer.

This results in:$f_\theta(x) = f_{\theta'}((x, 1))$,
where θ' is the parameter adpated in the above schema.

Usually proofs get a lot simple when you just have to wory about keeping one input constant. I think a lot of the proofs from the paper should still hold, for example Leamma 2.1 doesn't change.

For this to work, we need to add the additional requirement that $\sigma(1) = 1$

3 Proof

In the following section, neural networks with biases as defined as:
$f_\theta(x) = \sigma_{L+1}(\sigma_L(\dots\sigma_2(\sigma_1(xW^0 + b^0)W^1 + b^1)W^2 + b^2)\dots W^L + b^L)$.
We call b^i bias and additionally need the follwing constraint on the activation function σ_i:
$\forall i \in L + 1, \dots, 1.\exists c_i.\sigma_i(c_i) = 1$

Definition 3.1. Homogenous Coordinates for Neural Networks
Let f_θ be a neural network with biases. Then augumented neural network $f'_{\theta'}$ is a neural network as defined in Definition 1, (2.1) in [TODO: ref], where:
$\theta' = (W'^0, \dots, W'^L)$

and

$$W'^i = \begin{bmatrix} w_{11}^i & w_{12}^i & w_{13}^i & \dots & w_{1n}^i & bias_1^i \\ w_{21}^i & w_{22}^i & w_{23}^i & \dots & w_{2n}^i & bias_2^i \\ \dots & \dots & \dots & \dots & \dots & \dots \\ w_{d1}^i & w_{d2}^i & w_{d3}^i & \dots & w_{dn}^i & bias_n^i \\ 0 & \dots & \dots & \dots & 0 & c_i \end{bmatrix} = \begin{bmatrix} W^i & & & b^i \\ 0 & \dots & 0 & c_i \end{bmatrix} \in \mathbb{R}^{l_i+1,h_i+1} \forall i \in \{0, \dots L\}$$

$$W'^i = \begin{bmatrix} W^i & b^i \end{bmatrix} \in \mathbb{R}^{l_i+1,h_i} \text{ with } i = L+1$$

Theorem 3.1. *Equality of the augumented NN*
For all NN with Bias $f_{\theta'}$:
$f_{\theta'}(x) = f'_{\theta'}((x,1))$

Proof. via Induction over L.
For $L = 0$...
TODO, but should be pretty trivial □

References