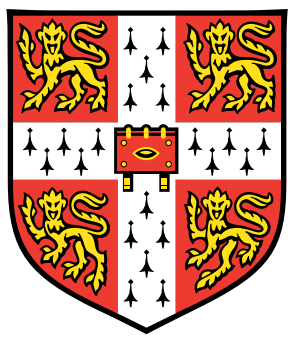
# Parameterised Sigmoid and ReLU Hidden Activation Functions for DNN Acoustic Modelling

Chao Zhang & Phil Woodland

Natural Speech Technology

Edinburgh – Cambridge – Sheffield

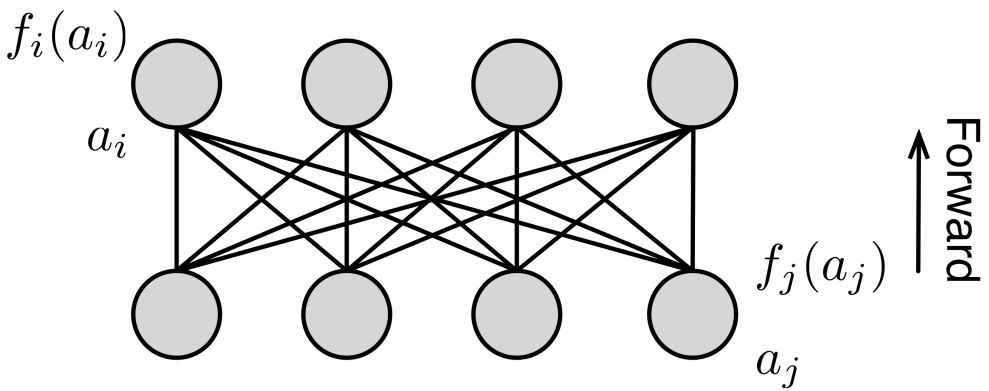


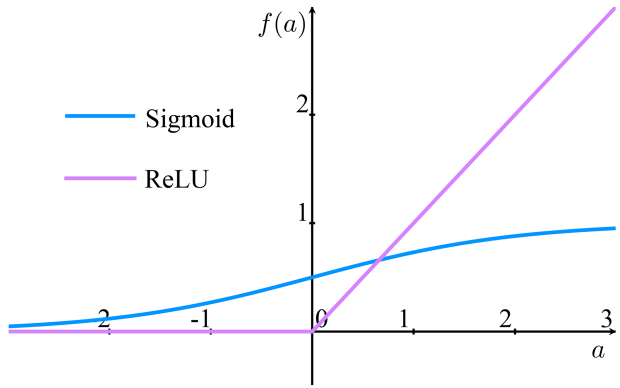
University of Cambridge

29 April 2015

# Introduction

- The hidden activation function plays an important role in deep learning: Pretraining (PT) & Finetuning (FT), ReLU.




- Recent studies on learning parameterised activation functions resulted in improved performance.
- We study the parameterised forms of Sigmoid ($p-$Sigmoid) and ReLU ($p-$ReLU) functions for SI DNN acoustic model training:

# Parameterised Sigmoid Function

- The generalised form of Sigmoid, or the *logistic function*, is

$$f_i(a_i) = \eta_i \cdot \frac{1}{1 + e^{-\gamma_i a_i + \theta_i}}$$

- $\eta_i$, $\gamma_i$, and $\theta_i$ have different effects on $f_i(a_i)$:
  - $\eta_i$ defines the boundaries of $f_i(a_i)$. It **L**earns **H**idden **U**nit $i$'s (positive, zero, or negative) **C**ontribution (LHUC).
  - $\gamma_i$ controls the steepness of the curve;
  - $\theta$ applies a horizontal displacement to $f_i(a_i)$.
- No bias term is added to $f_i(a_i)$, since it works the same as the bias of the layer.

# Parameterised Sigmoid Function

- By varying the parameters, $p-\text{Sigmoid}(\eta_i, \gamma_i, \theta_i)$ can do piecewise approximation to other functions, e.g., step, ReLU, and Soft ReLU.
- Can also present tanh, if the bias of the layer is taken into account.

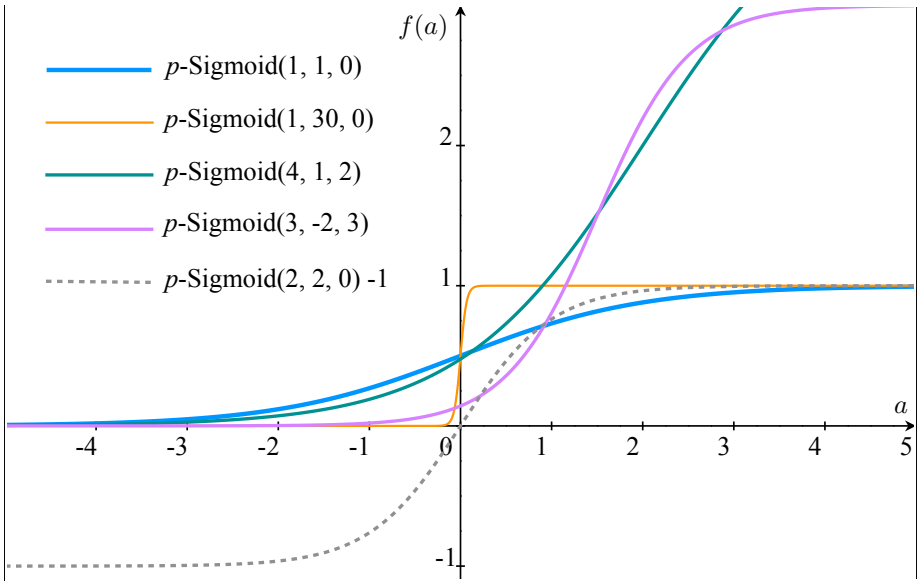



Figure: Piecewise approximation by $p-$Sigmoid functions.

# Parameterised ReLU Function

- Associate a scaling factor to either part of the function, to enable the 2 ends of the "hinge" rotate separately around the "pin".

$$f_i(a_i) = \begin{cases} \alpha_i \cdot a_i & \text{if } a_i > 0 \\ \beta_i \cdot a_i & \text{if } a_i \leqslant 0 \end{cases} .$$

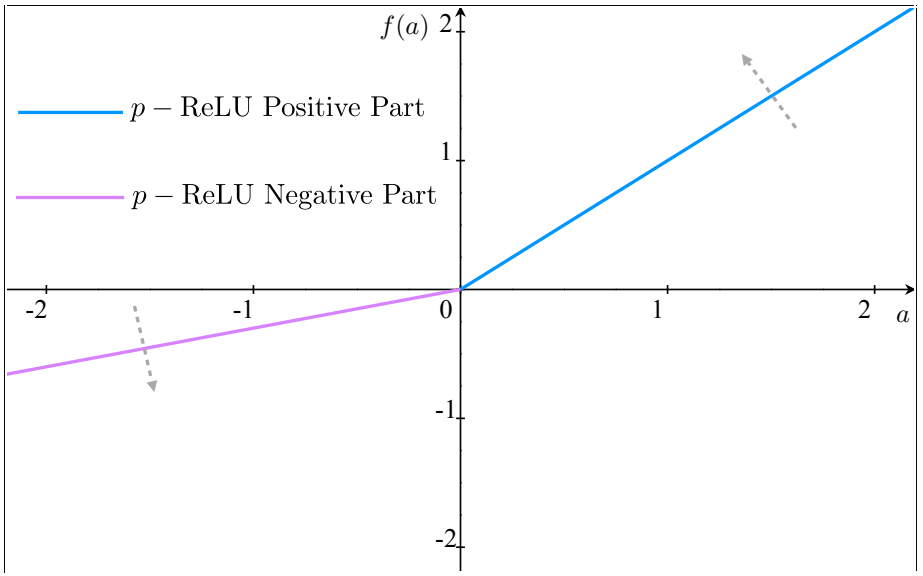



Figure: Illustration of the hinge-like shape $p-$ReLU function.

# EBP for Parameterised Activation Functions



- Assume
  - $\mathcal{F}$ is the objective function
  - $i$, $j$ are the output and input node numbers of a layer
  - $a_i$, $f_i(\cdot)$ are the activation value and activation function of node $i$
  - $\vartheta_i$ is a parameter of $f_i(\cdot)$, $w_{ji}$ is a (extended) weight
- According to the chain rule, there is

$$\frac{\partial \mathcal{F}}{\partial \vartheta_i} = \frac{\partial f_i(a_i)}{\partial \vartheta_i} \sum_j \frac{\partial \mathcal{F}}{\partial a_j} w_{ji}.$$

- Therefore, we need to compute
  - $\partial f_i(a_i)/\partial a_i$ for training weights & biases;
  - $\partial f_i(a_i)/\partial \vartheta_i$ for activation function parameter $\vartheta_i$.

# Experiments

- CE DNN-HMMs were trained on 72 hours Mandarin CTS data.
- Three testing sets were used, *dev04*, *eval03*, and *eval97*.
  - 42d CMLLR(HLDA(PLP_0_D_A_T_Z)+Pitch_D_A) features.
  - Context shift set, $\mathbf{c} = [-4, -3, -2, -1, 0, +1, +2, +3, +4]$.
- 63k word dictionary and trigram LM trained using 1 billion words.
- DNN structure $378 \times 1000^5 \times 6005$.
- Imporved *NewBob* learning rate scheduler
  - Sigmoid & $p-$Sigmoid: $\tau_0 = 2.0 \times 10^{-3}$, $N_{\text{min}} = 12$
  - ReLU & $p-$ReLU: $\tau_0 = 5.0 \times 10^{-4}$, $N_{\text{min}} = 8$
- $\eta_i$, $\gamma_i$, $\theta_i$, $\alpha_i$, and $\beta_i$ are intialised as 1.0, 1.0, 0.0, 1.0, and 0.25.
- All GMM-HMM and DNN-HMM acoustic model training and decoding used HTK.

# Experiments with $p-$Sigmoid

- Learning $\eta_i$, $\gamma_i$, and $\theta_i$ in PT & FT, and FT only.
- Using combinations did not outperform using separate parameters.

| ID | Activation Function | dev04 |
|---|---|---|
| S0 | Sigmoid | 27.9 |
| S1$^+$ | $p$-Sigmoid$(\eta_i, 1, 0)$ | 27.6 |
| S2$^+$ | $p$-Sigmoid$(1, \gamma_i, 0)$ | 27.7 |
| S3$^+$ | $p$-Sigmoid$(1, 1, \theta_i)$ | 27.7 |
| S1 | $p$-Sigmoid$(\eta_i, 1, 0)$ | 27.1 |
| S2 | $p$-Sigmoid$(1, \gamma_i, 0)$ | 27.5 |
| S3 | $p$-Sigmoid$(1, 1, \theta_i)$ | 27.4 |
| S6 | $p$-Sigmoid$(\eta_i, \gamma_i, \theta_i)$ | 27.3 |

Table: *dev04 %WER for the p-Sigmoid systems. $^+$ means the activation function parameters were trained in both PT and FT.*

# Experiments with $p-$ReLU

- For ReLU DNNs, it is not useful to avoid the impact on the other parameters at the begining.
- $\alpha_i$ has more impact on training than $\beta_i$.

| ID | Activation Function | dev04 |
|---|---|---|
| R0 | ReLU | 27.6 |
| R1 | $p$-ReLU$(\alpha_i, 0)$ | 26.8 |
| R2 | $p$-ReLU$(1, \beta_i)$ | 27.0 |
| R3 | $p$-ReLU$(\alpha_i, \beta_i)$ | 27.1 |
| R1$^-$ | $p$-ReLU$(\alpha_i, 0)$ | 27.4 |
| R2$^-$ | $p$-ReLU$(1, \beta_i)$ | 27.0 |

Table: *dev04 %WER for the p-ReLU systems. $^-$ indicates the activation function parameters were frozen in the 1st epoch.*

# Results on All Testing Sets

- S1 and R1 had 3.4% and 2.0% lower WER than S0 and R0, by increasing only 0.06% parameters.
- *p*-Sigmoid contains gains by making Sigmoid similar to ReLU.
- Weighting the contribution of each hidden unit individually is quite useful.

| ID | Activation Function | eval97 | eval03 | dev04 |
|---|---|---|---|---|
| S0 | Sigmoid | 34.1 | 29.7 | 27.9 |
| S1 | *p*-Sigmoid($\eta_i, 1, 0$) | 32.9 | 28.6 | 27.1 |
| R0 | ReLU | 33.3 | 29.1 | 27.6 |
| R1 | *p*-ReLU($\alpha_i, 0$) | 32.7 | 28.7 | 26.8 |

Table: *%WER on all test sets.*

# Summary

- Different types of parameters and their combinations of $p-$Sigmoid and $p-$ReLU were analysed and compared.
- A scaling factor with no constraint imposed is the most useful. With the linear scaling factors,
  - $p-$Sigmoid$(\eta_i, 1, 0)$ resulted 3.4% relative WER reduction compared with Sigmoid.
  - $p-$ReLU$(\alpha_i, 0)$ reduced the WER by 2.0% relative over the ReLU baseline.
- Learning different types of parameters simultaneously is difficult.

# Appendix: Parameterised Sigmoid Function



- To compute the exact derivatives, it is necessary to store $a_i$.

$$\frac{\partial f_i(a_i)}{\partial a_i} = \begin{cases} 0 & \text{if } \eta_i = 0 \\ \gamma_i f_i(a_i)\left(1 - \eta_i^{-1} f_i(a_i)\right) & \text{if } \eta_i \neq 0 \end{cases},$$

$$\frac{\partial f_i(a_i)}{\partial \eta_i} = \begin{cases} \left(1 + e^{-\gamma_i a_i + \theta_i}\right)^{-1} & \text{if } \eta_i = 0 \\ \eta_i^{-1} f_i(a_i) & \text{if } \eta_i \neq 0 \end{cases},$$

$$\frac{\partial f_i(a_i)}{\partial \gamma_i} = \begin{cases} 0 & \text{if } \eta_i = 0 \\ a_i f_i(a_i)\left(1 - \eta_i^{-1} f_i(a_i)\right) & \text{if } \eta_i \neq 0 \end{cases},$$

$$\frac{\partial f_i(a_i)}{\partial \theta_i} = \begin{cases} 0 & \text{if } \eta_i = 0 \\ -f_i(a_i)\left(1 - \eta_i^{-1} f_i(a_i)\right) & \text{if } \eta_i \neq 0 \end{cases},$$

- For $p-$Sigmoid$(\eta_i, 1, 0)$, let $\partial f_i(a_i)/\partial \eta_i$ $(\eta_i = 0)$, $a_i$ is not necessary to be stored.

# Appendix: Parameterised ReLU Function

- Since $\alpha_i$, $\beta_i$ can be any real number, it is not possible to tell the sign of $a_i$ from $f_i(a_i)$.

$$\frac{\partial f_i(a_i)}{\partial a_i} = \begin{cases} \alpha_i & \text{if } a_i > 0 \\ \beta_i & \text{if } a_i \leqslant 0 \end{cases},$$

$$\frac{\partial f_i(a_i)}{\partial \alpha_i} = \begin{cases} a_i & \text{if } a_i > 0 \\ 0 & \text{if } a_i \leqslant 0 \end{cases},$$

$$\frac{\partial f_i(a_i)}{\partial \beta_i} = \begin{cases} 0 & \text{if } a_i > 0 \\ a_i & \text{if } a_i \leqslant 0 \end{cases}.$$

- For $p-\text{ReLU}(\alpha_i, 0)$, let $\partial f_i(a_i)/\partial \alpha_i$ $(\alpha_i = 0)$, it is not necessary to store $a_i$.