ON THE EQUIVALENCE BETWEEN PREDICTIVE MODELS FOR AUTOMATIC SPEECH RECOGNITION

Bojan Petek
University of Ljubljana
Faculty of Natural Sciences and Engineering
Snežniška 5, 1000 Ljubljana, SLOVENIA
e-mail: bojan.petek@uni-lj.si

ABSTRACT

This paper addresses the equivalence of input-output mapping functions between the Linked Predictive Neural Networks (LPNN, [5]) and the Hidden Control Neural Networks (HCNN, [2]). Two novel theoretical results supported with *Mathematica* experiments are presented. First, it is proved that for every HCNN model there exist an equivalent LPNN model. Second, it is shown that the set of input-output functions of an LPNN model is strictly larger than the set of functions of an equivalent HCNN model. Therefore, when using equal architecture of the canonical building blocks (MLPs) for the LPNN and HCNN models, the LPNN represent a superset of the approximation capabilities of the HCNN models.

1 INTRODUCTION

Predictive neural networks [1, 5, 2] represent non-linear dynamical system models generalized to the segment framework [3]. Their performance in automatic speech recognition was found to be inferior to that of the advanced HMMs, mainly due to problems with identity mappings in modeling intrasegmental dependencies [6, 4]. It is belived, however, that their strenghts as non-linear models could still be exploited in modeling dependencies across phones, e.g. diphones, where linear models prove inadequate. Therefore, studies of approximation capabilities of nonlinear models are important given their potential use in modeling the coarticulation effect across phones.

2 COMPARING LPNN WITH HCNN

This section gives a theoretical comparison between the LPNN and HCNN models [5, 2]. Following formal definitions of the models, their approximation capabilities are compared and discussed.

2.1 Model Definitions

Following [5], the LPNN model is defined as a sequence of S distinct MLPs, where each network is associated with particular state $c, 1 \leq c \leq S$. [1] Approximation capabilities of the LPNN model can be described as

$$\begin{aligned} \mathcal{F}_{LPNN} &= \text{LPNN } (S; w_1, v_1, \theta_1; \cdots; w_S, v_S, \theta_S) \\ &= \left\{(f_1, \cdots, f_S); f_c : \mathbf{R}^{N_i} \rightarrow \mathbf{R}\right\} \end{aligned} \quad (1)$$

where w_i denote weights from the input to hidden units, v_i weights from the hidden to output units, θ_i bias weights and

$$f_c(x) = \sum_{j=1}^{N_h} v_{cj}\sigma(w_c^T x + \theta_{cj}) \quad (2)$$

represents a mapping function of a canonical MLP with N_i input, N_h hidden units that have sigmoidal activation function σ.

Note that for each state c there is a separate network which realizes the set of functions $\mathcal{F}_{N_i}^{N_h}(\sigma)$ given by

$$\mathcal{F}_{N_i}^{N_h}(\sigma) = \left\{ f : \mathbf{R}^{N_i} \rightarrow \mathbf{R} \mid f(x) = \sum_{j=1}^{N_h} v_j \sigma(w^T x + \theta_j) \right\} \quad (3)$$

where σ is the activation function of the hidden units and T denotes a transpose. Therefore, given S states, the LPNN model realizes an S-tuple of functions with S independent parameter sets

$$\{w_i, v_i, \theta_i; 1 \leq i \leq S\} \quad (4)$$

The independence of parameter sets means that no LPNN parameters need to be shared, i.e., need to be set equal, in any of two states of the model.

An HCNN model definition [2] is given in Figure 1. By rewriting the Equation 3 for the HCNN model, we get

$$\begin{aligned} \mathcal{F}_{HCNN} &= \text{HCNN } (N_c; w, v, \theta, \omega) \\ &= \left\{(f_1, \cdots, f_{N_c}); f_c : \mathbf{R}^{N_i} \rightarrow \mathbf{R}\right\} \end{aligned} \quad (5)$$

where

$$f_c(x) = \sum_{j=1}^{N_h} v_j \sigma(w^T x + \omega^T c + \theta_j) \quad . \quad (6)$$

[1] For sake of notational convenience assume one output unit, $N_o = 1$.

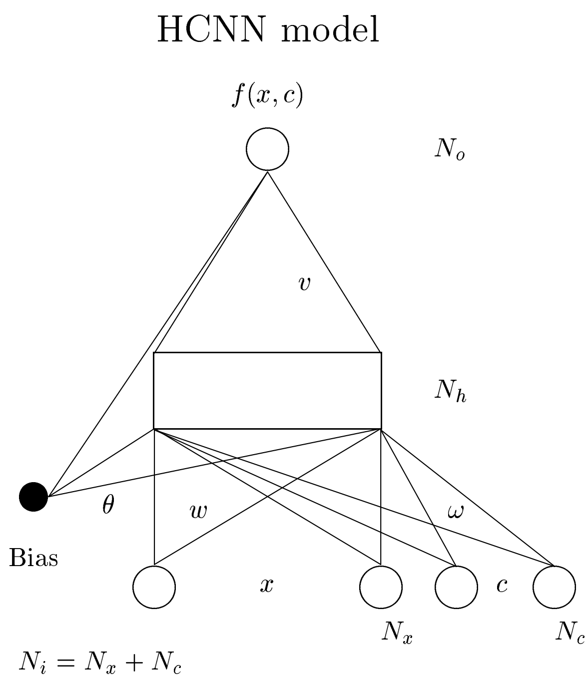

Figure 1: An HCNN model.

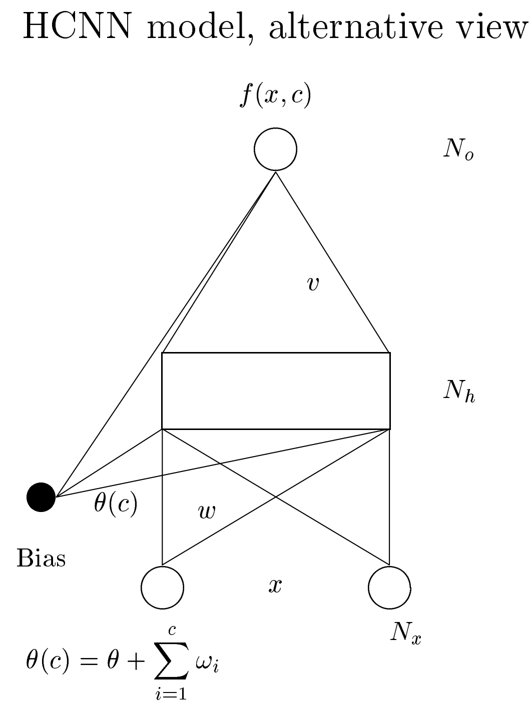

Figure 2: An alternative view of the HCNN model.

where ω denotes the weights from hidden control inputs to the first hidden layer and c the hidden control input vector. Note that the values of c are restricted to be of "thermomether" type, i.e.,

$$c_i = \underbrace{1111}_{c}\overbrace{00000}^{N_c} \tag{7}$$

This means that given any state c, $1 \leq c \leq N_c$, the first c consecutive inputs are set to 1.0 whereas the remaining $N_c - c$ inputs are set to zero.

In contrast to the LPNN model, the HCNN model consists of a single MLP. This means that the model parameters w, v, θ are shared among all states. To enable better modeling of sequence of states, an additional hidden control input (HCI) modulates the function f_c for states $c, 1 \leq c \leq S$, where $N_c = S$.

An equivalent view of the HCNN model is given in Figure 2.

When considering activation levels of the HCI inputs (i.e., thermometer representation), we may group the connections from the HCI to the hidden layer as additional bias weights, i.e.,

$$\theta(c) = \theta + \sum_{i=1}^{c} \omega_i \tag{8}$$

where θ denotes bias weights, and ω_i weights from the HCI to hidden units for a particular state c.

2.2 From HCNN to LPNN model

Considering definitions of the LPNN and HCNN models, the following theorem can be proved.

Theorem 1 *For any HCNN model there exists an equivalent LPNN model.*

Proof: Consider an HCNN model with N_x input units, N_c hidden control inputs, N_h hidden units on the first hidden layer, and N_o output units. Let the set of HCNN parameters (i.e., weights) be denoted by w_0, for weights from input x to the first hidden layer, ω, for weights from HCI to the first hidden layer, v_0, for weights from first hidden layer to the outputs, and θ_0, for bias weights. Then an equivalent LPNN model has N_x input units, N_h hidden units on the first hidden layer, and N_o output units. Let $w_i, 1 \leq i \leq N_c$ denote weights of the LPNN's i-th MLP, connecting input x to the first hidden layer, $v_i, 1 \leq i \leq N_c$ denote connections from first hidden layer to output units, and $\theta_i, 1 \leq i \leq N_c$ denote bias weights.

According to Equation 8 (Figure 2), the parameters of an equivalent LPNN model are defined by

$$\begin{aligned} S &= N_c \\ w_i &= w_0, 1 \leq i \leq N_c \\ v_i &= v_0, 1 \leq i \leq N_c \\ \theta_i &= \theta_0 + \sum_{i=1}^{c} \omega_i \end{aligned} \tag{9}$$

The Equations 9 impose

$$\begin{aligned} &\text{HCNN } (N_c; w, v, \theta, \omega) = \\ &\text{LPNN } (N_c; w, v, \theta + \omega_1; \\ &\qquad w, v, \theta + \omega_1 + \omega_2; \\ &\qquad \cdots, \\ &\qquad w, v, \theta + \textstyle\sum_{j=1}^{N_c} \omega_j) \end{aligned} \tag{10}$$

thereby

$$\mathcal{F}_{LPNN} = \mathcal{F}_{HCNN} \tag{11}$$

as defined by Equations 1 and 5.

□

2.3 From LPNN to HCNN model

The following proposition describes a transition from an LPNN to the HCNN model.

Proposition 2 *The set of functions of an LPNN model*

$$\mathcal{F}_{LPNN} = \text{LPNN}\ (S; w_1, v_1, \theta_1; \cdots; w_S, v_S, \theta_S)$$
$$= \left\{ (f_1, \cdots, f_S); f_c : \mathbf{R}^{N_i} \to \mathbf{R} \mid f_c(x) = \sum_{j=1}^{N_h} v_{cj}\sigma(w_c^T x + \theta_{cj}) \right\}$$

is strictly larger than the set of functions

$$\mathcal{F}_{HCNN} = \text{HCNN}\ (N_c; w, v, \theta, \omega)$$
$$= \left\{ (f_1, \cdots, f_{N_c}); f_c : \mathbf{R}^{N_i} \to \mathbf{R} \mid f_c(x) = \sum_{j=1}^{N_h} v_j\sigma(w^T x + \omega^T c + \theta_j) \right\}$$

of an equivalent HCNN model.

Proof: Consider a simple LPNN model with $N_i = N_h = 1, S = 2$ that realizes the functions of the form (a superscript denotes the state $c, 1 \le c \le S$, ie, the LPNN network index)

$$\begin{array}{c} \sigma(\theta_2^1, v_1^1; \sigma(\theta_1^1, w_1^1; x)) \\ \sigma(\theta_2^2, v_1^2; \sigma(\theta_1^2, w_1^2; x)) \end{array} \tag{12}$$

whereas an equivalent HCNN model realizes

$$\begin{array}{c} \sigma(\theta_2^0, v_1^0; \sigma(\theta_1^0 + \omega_1, w_1^0; x)) \\ \sigma(\theta_2^0, v_1^0; \sigma(\theta_1^0 + \omega_1 + \omega_2, w_1^0; x)) \end{array} \tag{13}$$

Since the weights w_1^0, v_1^0, θ_i^0 are shared (i.e., have to be equal) for different states c whereas LPNN model weights $w_1^c, v_1^c, \theta_i^c; c = 1, 2$ are not, the set of LPNN functions is strictly larger than that of the HCNN model.

□

To illustrate the results discussed above with models having two-dimensional inputs, consider an HCNN model with two inputs as depicted in Figure 3.

According to Equation 8, the HCI weights can be seen as changing the value of the bias weight on the first hidden layer, $\theta_1^0(c)$ (Figure 3). The effects of this change on the function $f_0(x_1, x_2, c)$ for three discrete values of $\theta_1^0(1) = -10, \theta_1^0(2) = 0, \theta_1^0(3) = 10$ are shown in Figure 4. The values of other weights of the model have been set to $w_1^0 = 5, w_2^0 = -5, \theta_2^0 = 2, v_1^0 = -10$. Figure 3 clearly shows that changing the weight $\theta_1^0(c)$, that is, switching the state of the HCNN model, has a *limited* influence on the function of the model, $f_0(x_1, x_2, c)$. Roughly speaking, switching the state in the HCNN model results in a shifted version of its function of the previous state.

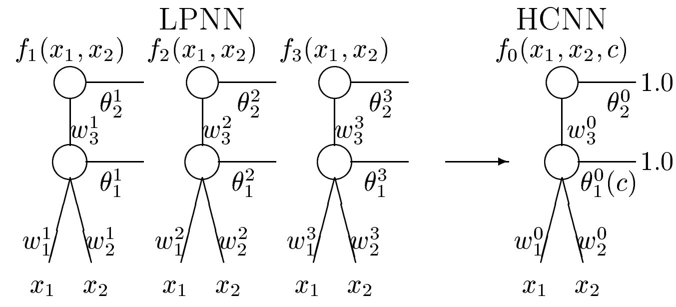

Figure 3: From two input LPNN to equivalent HCNN model (see text).

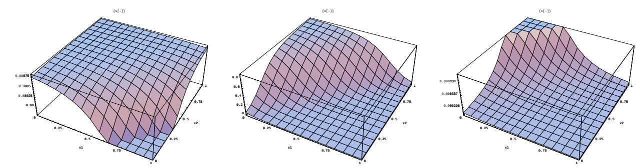

Figure 4: Input-output functions of the HCNN with two inputs for three discrete values of $\theta_1^0(c)$.

Consider now an LPNN model realizing the functions shown in Figure 5. The weight values of the LPNN model have been set to $\theta_1^1 = -1, w_1^1 = -6, w_2^1 = 5, \theta_2^1 = 2, v_1^1 = -1$ for the first MLP, $\theta_1^2 = 5, w_1^2 = -5, w_2^2 = -6, \theta_2^2 = 2, v_1^2 = -10$ for the second MLP, and $\theta_1^3 = -10, w_1^3 = -5, w_2^3 = -3, \theta_2^3 = -6, v_1^3 = -10$ for the last MLP. Clearly the HCNN model *cannot* uniquely replace the LPNN model because changing the state in the HCNN model results in a shifted version of its function (Figures 6 to 8).

Figure 6 (left) shows the functions of the HCNN model with weight values set to $\theta_1^0 = -1, w_1^0 = -6, w_2^0 = 5, \theta_2^0 = 2, v_1^0 = -1$. Clearly, equality of the LPNN (*state*$_1$) and HCNN (*state*$_1$) weights implies $f_1(x_1, x_2) = f_0(x_1, x_2, 1), \quad \forall x$, i.e., identical functions of both models. But switching the state in the HCNN model with $\theta_1^0 = 10$ produces only the shifted version of the $f_0(x_1, x_2, 1)$ (Figure 6, center). Similar result with $\theta_1^0 = -10$ is shown in the Figure 6 (right).

Figure 7 (left) shows the functions of the HCNN model with weight vector of $\theta_1^0 = 5, w_1^0 = -5, w_2^0 = -6, \theta_2^0 = 2, v_1^0 = -10$. The equality of the LPNN (*state*$_2$) and HCNN (*state*$_2$) weights implies $f_2(x_1, x_2) = f_0(x_1, x_2, 2), \quad \forall x$, i.e., identical functions of both models. But switching the state in the HCNN model with $\theta_1^0 = 20$ produces only the shifted version of $f_0(x_1, x_2, 2)$ (Figure 7, center). Similar result with $\theta_1^0 = -20$ is shown in the Figure 7 (right).

Finally, Figure 8 (left) shows the functions of the HCNN model with weight values set to $\theta_1^0 = -10, w_1^0 = -5, w_2^0 = -3, \theta_2^0 = -6, v_1^0 = 10$. Again the equality of the LPNN (*state*$_3$) and HCNN (*state*$_3$) weights implies

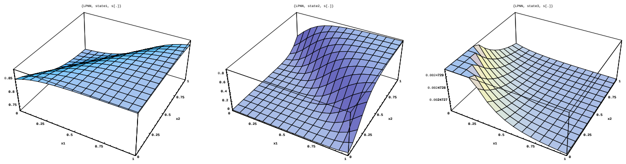

Figure 5: Input-output functions of a three state LPNN model with two dimensional inputs.

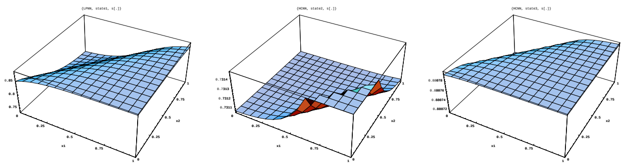

Figure 6: Approximating the first state function of the LPNN model with an HCNN model.

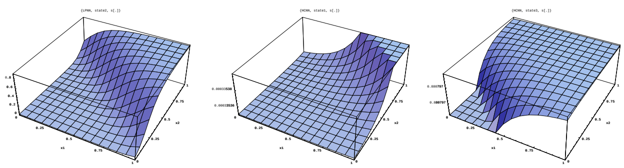

Figure 7: Approximating the second state function of the LPNN model with an HCNN model.

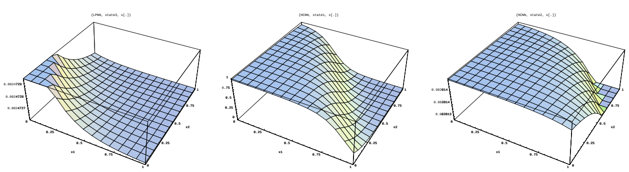

Figure 8: Approximating the third state function of the LPNN model with an HCNN model.

$f_3(x_1, x_2) = f_0(x_1, x_2, 3), \quad \forall x$, i.e., identical functions. But switching the state in the HCNN model with $\theta_1^0 = 5$ produces only the shifted version of the $f_0(x_1, x_2, 3)$ (Figure 8, center). Similar result with $\theta_1^0 = 20$ is shown in the Figure 8 (right).

3 CONCLUSIONS

The paper has considered the equivalence of input-output mapping functions between the Linked Predictive Neural Networks (LPNN, [5]) and the Hidden Control Neural Networks (HCNN, [2]).

It is proved that for every HCNN model there exist an equivalent LPNN model. In other words, for every HCNN there exist an equivalent LPNN with exactly the same input-output mapping functions ($\mathcal{F}_{LPNN} = \mathcal{F}_{HCNN}$, as defined by Equations 1 and 5).

It is also shown and supported by the *Mathematica* experiments that the set of input-output functions of an LPNN model is strictly larger than the set of functions of an equivalent HCNN model. Despite the simplicity of analyzed models the Figures 4 to 8 give an insight into the role of each of the model parameters on the mapping function. Specifically, given the fact that the state change in the HCNN model is equivalent to a modulation of the weight value $\theta(c)$ (Equation 8), it is clear that the set of functions of the LPNN model $\mathcal{F}_{LPNN}$ is strictly larger than the set of functions of the equivalent HCNN model $\mathcal{F}_{HCNN}$.

In other words, the LPNN model can realize mappings which the equivalent HCNN model cannot, given the same structure of the multi-layer perceptron as a canonical building block (ie, the number of input, hidden, output units; the number of weights).

An interesting further research avenue might be to compare the mapping functions ($\mathcal{F}_{LPNN}$ and $\mathcal{F}_{HCNN}$) that the learning algorithms actually yield on the same real-world task.

References

[1] K. Iso, "Speech Recognition Using Neural Prediction Model", *IEICE Technical Report SP89-23*, pp. 81-87, 1989.

[2] E. Levin, "Word Recognition Using Hidden Control Neural Architecture", *Proc. IEEE Int. Conf. on ASSP*, pp. 433-436, April 1990.

[3] M. Ostendorf, V. Digalakis, and O. Kimball, "From HMMs to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition," Boston University Technical Report No. ECS-95-002.

[4] B. Petek, A. H. Waibel, and J. M. Tebelskis, "Integrated and Phoneme-Function Word Architecture of Hidden Control Neural Networks for Continuous Speech Recognition", *Speech Communication*, Special Issue on Eurospeech'91, Vol. 11, Nos. 2-3, pp. 273-282, June 1992.

[5] J. Tebelskis, and A. Waibel, "Large Vocabulary Recognition Using Linked Predictive Neural Networks", *Proc. IEEE Int. Conf. on ASSP*, pp. 437-440, April 1990.

[6] J. Tebelskis, A. Waibel, B. Petek and O. Schmidbauer, "Continuous Speech Recognition Using Linked Predictive Neural Networks", *Proc. IEEE Int. Conf. on ASSP*, pp. 61-64, May 1991.