Redes neuronales convolucionales y redes neuronales recurrentes en la transcripción automática

Research · July 2019

DOI: 10.13140/RG.2.2.10855.39843

1 author:

Juan Sebastian Gelvez Prieto
National University of Colombia
3 PUBLICATIONS **1** CITATION

SEE PROFILE

Redes neuronales convolucionales y redes neuronales recurrentes en la transcripción automática

Juan Sebastian Gelvez Prieto
Estudiante de lingüística
Universidad Nacional de Colombia
jsgelvezp@unal.edu.co

Resumen

Todo lingüista ha tenido que enfrentarse a la tarea de transcribir, si bien no necesariamente el corpus extenso de una lengua, al menos unas cuantas oraciones en su lengua madre. El problema es que las transcripciones van a variar de persona a persona, dependiendo de su experiencia y habilidad, siendo unas más acertadas que otras. La forma en que se ha buscado solución a este problema es con el uso de redes neuronales, específicamente redes neuronales convolucionales y redes neuronales recurrentes. Con el uso de ambos tipos de redes neuronales se propone en este artículo una metodología mixta que permita automatizar el proceso de transcripción de una lengua. En esta investigación se encontró que la forma en que se pueden combinar estas dos arquitecturas es dejando a las redes neuronales convolucionales extraer características acústicas de alto nivel, mientras que a las redes neuronales recurrentes se les asigna la tarea de clasificar secuencias. Así se reducirá la posibilidad de error en las transcripciones eliminando el factor subjetivo que subyace al trabajo hecho por humanos.

Palabras clave: redes neuronales convolucionales, redes neuronales recurrentes, transcripción automática, deep learning

1. Introducción

La transcripción es una tarea a la cual todos los lingüistas se han enfrentado alguna vez en la vida. El proceso que se lleva a cabo a la hora de plasmar de manera escrita una lengua que no ha sido documentada anteriormente, requiere de bastantes horas de trabajo si se hace manualmente, inclusive después de que ya se ha recolectado la información, que es un proceso que lleva también bastante tiempo, además los resultados obtenidos pueden ser poco precisos, por lo que se depende de un profesional que tenga mucha experiencia en este campo.

La conversión de habla a texto ya se ha investigado anteriormente y se han plasmado los resultados en una gran cantidad de artículos y eventos, como *Text, Speech and Dialogue* llevado a cabo todos los años en la República Checa, en el que se hacen conferencias hablando de los resultados de investigaciones dentro de cada uno de los ejes principales del evento. Inclusive, la conversión de habla a texto tiene numerosas aplicaciones tanto en investigación como en la vida cotidiana, como por ejemplo la herramienta que tiene *Google* para poder escribir documentos utilizando la voz. Es común, dentro de los artículos recientes, el uso del *deep learning* utilizando redes neuronales artificiales, entre las que destacan los dos tipos que se presentan en el presente documento: redes neuronales convolucionales y redes neuronales recurrentes. Tóth (2015) mencionó en su investigación que estaba buscando combinar estos dos tipos de redes neuronales (NNs, por sus siglas en inglés) para hacer transcripción automática. Basado en esto, el objetivo de este artículo es realizar una revisión de literatura reciente que permita desarrollar una metodología para la transcripción automática que combine los dos tipos de redes neuronales arriba mencionados. Cabe aclarar que este artículo se centrará en una explicación de los conceptos de una manera no matemática, de forma que pueda ser entendido por personas no expertas en esta ciencia, aunque sí se debe tener conocimiento previo en el campo de la informática para una mejor comprensión.

Este artículo está dividido de la siguiente manera: en la sección 2 se describe la metodología utilizada; en la sección 3 se explica lo que son las redes neuronales convolucionales y todo lo referente a estas; en el apartado 4 se describe las redes neuronales recurrentes y lo que con estas se relaciona; en la sección 5 se encuentran la discusión, la conclusión y opciones para futuros trabajos.

2. Metodología

Con el objetivo de que el presente artículo recupere las investigaciones más recientes sobre el tema se utilizó el motor de búsqueda *Google Académico*, en el cual se introdujeron las palabras clave 'automatic phone recognition AND CNN' y 'automatic phone recognition AND RNN', además de un filtro de tiempo que tomó los artículos de investigación que fueran publicados después del año 2010, para la posterior recolección de los artículos de investigación que cumplieran con estos criterios. En adición, se tomaron artículos que fueran citados por los que se recolectaron anteriormente y que resultaron útiles, que además cumplieran con los requisitos antes mencionados.

3. Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNNs, por sus siglas en inglés) son un tipo de redes neuronales artificales que son utilizadas, entre otras cosas, para el análisis y reconocimiento de imágenes. Khan et al. dicen que la forma en que una CNN trabaja es muy parecida a como lo hacen las redes neuronales convencionales pero se diferencian en un aspecto, "[...] each neuron in a CNN layer is a two- (or high-) dimensional filter which is convolved with the input of that layer"(2018), lo que resulta fundamental al querer tratar con el aprendizaje de patrones como los de una imagen. Dentro de este tipo de NNs, las capas convolucionales son la parte más importante de toda la red. Estas capas, como lo mencionan Khan et al. (2018), son un grupo de filtros (para ver un ejemplo de los filtros de una primera capa véase la figura 1) que son convolucionados con una cierta entrada (o *input*) para generar un mapa de características en una salida (o *output*). Las CNNs toman un dato, por ejemplo una imagen, y aplican los filtros, no a toda la imagen, sino a una parte de esta, luego hacen lo mismo en otro sector, y así sucesivamente hasta obtener algo más fácil de procesar. Las primeras capas de estas NNs se encargan de identificar los patrones más básicos, como puede ser una línea recta vertical, o cierta curva, y a medida que cada capa pasa su *output* a la siguiente, se van identificando patrones cada vez más complejos a partir de los anteriores y así hasta llegar a identificar cosas más abstractas, como puede ser un gato en una imagen.

En el reconocimiento del habla, las CNNs han demostrado un rendimiento superior al de una red neuronal profunda (DNN) estándar, como es el caso descrito por Abdel-Hamid et al. (2013) en el que una CNN con *full weight sharing* tuvo una reducción del 5 % en la tasa de error de fonema (PER, por sus siglas en inglés) en la prueba TIMIT de reconocimiento de fonemas, en comparación con una DNN sin convolución. Otro caso en el que una CNN supera en desempeño a una DNN es en la convolución en el dominio de la frecuencia (Abdel-Hamid et al., 2012; Deng et al., 2013; Sainath et al., 2013b; Tóth, 2014), o en el reconocimiento de grandes cantidades de vocabulario (LVCSR, por sus siglas en inglés), donde obtuvieron de 12 % a 14 % de mejora en la tasa de error de palabra frente a las DNNs simples, como lo menciona Tóth (2015) parafraseando a Sainath et al. (2015).

La forma en que las CNNs operan difiere de otras DNNs en tres aspectos que menciona Tóth (2015) citando a Abdel-Hamid et al. (2012) los cuales llaman *locality*, *weight sharing* y *pooling*. En este apartado se explicará el primero, debido a que los demás ya tienen una subsección propia dentro de esta sección, 3.3 y 3.2, respectivamente. A lo que *locality* se refiere es que una neurona convolucional procesa solamente una pequeña parte de toda la imagen, por lo que es más fácil de encontrar patrones en esta. La arquitectura de una CNN consiste básicamente de tres tipos de capas: convolucionales, de *pooling* y unas totalmente conectadas. Las neuronas que constituyen la primera capa son las encargadas de aplicar los filtros al *input* para de esta forma hacer más sencillo el aprendizaje de patrones, ver figura 1. Luego este resultado pasa a la capa de *pooling*, cuyo funcionamiento se explicará más adelante, y finalmente a las capas totalmente conectadas (o *fully-connected layers*) que funcionan como cualquier otra DNN normal.

3.1. Eje temporal

Antes, las CNNs habían sido utilizadas únicamente para el análisis del habla en el eje de la frecuencia, ya que en el eje del tiempo ya existía un modelo que podía hacer el análisis, el modelo oculto de Markóv (HMM, por sus siglas en inglés) (Abdel-Hamid et al., 2012; Deng et al., 2013; Sainath et al., 2013b). Pero debido a que puede haber pequeñas variaciones en los fonemas a lo largo del tiempo se decidió utilizar la convolución en

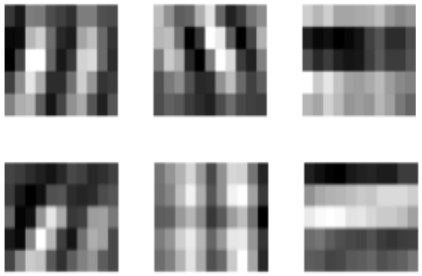

Figura 1: Filtros de una CNN. Imagen tomada de Huang et al. (2015)

el eje temporal también. Tanto en Abdel-Hamid et al. (2013) como en Tóth (2014) se encontró que la combinación de convolución en el eje temporal y en el eje de frecuencia produjo mejores resultados en el reconocimiento del habla, siendo la tasa de error del segundo de un 16.7 % en la prueba TIMIT, que fue un récord con ese grupo de datos.

3.2. Pooling

La operación *pooling*[1] es utilizada en las CNNs para hacer a estas redes más tolerantes a los pequeños cambios en frecuencia y tiempo. En esta operación las neuronas toman su *input* que puede ser un vector con diferentes vectores en su interior, y lo transforman en un único valor que será el *output* de esta neurona. En palabras de Abdel-Hamid et al. (2013): "A pooling layer is added on top of the convolution layer to compute a lower resolution representation of the convolution layer activations through sub-sampling". Una de estas operaciónes es Max-pooling que básicamente lo que hace es tomar el valor máximo del conjunto de valores de su input y lo pasa como output. Esta operación al igual que la función de activación maxout, que se verá adelante, tiene un pequeño inconveniente y es que se ha observado que tiende a sobreadapatarse a los datos (fenómeno conocido como *overfitting*). Para solucionar esto se ha propuesto utilizar otras funciones como puede ser el *pooling* estocástico, pero no se han observado mejores resultados (Sainath et al., 2013a). Por otro lado, también se ha utilizado la operación *p*-norm *pooling*, que en Zhang et al. (2014) demostró disminuir el *overfitting*. Sin embargo, ninguna de las dos anteriores demostró ser significativamente mejor que la operación de Max-pooling, según confirma Tóth (2015).

[1]Se utiliza este término debido a la carencia de uno equivalente en el español

3.3. Weight sharing

Cada neurona convolucional recibe un *input* que representa caraterísticas de un rango de frecuencia limitado. Por lo que neuronas que pertenezcan al mismo mapa de características (*feature map*) compartiran pesos (también llamados filtros o *kernels*), pero reciben diferentes *inputs* que varían en frecuencia (Abdel-Hamid et al., 2013). Esto es a lo que se llama *weight sharing* y hay dos tipos de este, el primero es el *full weight sharing* que es utilizado en el reconocimiento de imágenes ya que un solo patrón puede aparecer en diferentes partes de una imagen, mientras que en el análisis del habla los diferentes patrones aparecen en diferentes bandas de frecuencia, por lo que no es necesario utilizar un *total weight sharing* y se utiliza el *limited weight sharing*. Es decir, cada filtro se aplica a diferentes bandas de determinada frecuencia, por lo que utilizando este último *weight sharing* se logra disminuir el número de neuronas en la capa encargada del *pooling* lo que permite identificar patrones en diferentes bandas. Abdel-Hamid et al. (2013) sugieren que hay que agregar la capa de convolución de *limited weight sharing* encima (o en el tope) de una capa de *full weight sharing*, ya que un problema que presenta la primera es que no permite la adición de otras capas convolucionales encima.

3.4. Activación maxout

Las neuronas de una red neuronal se ponen en marcha gracias a una función de activación, básicamente lo que hacen estas funciones es convertir un *input* en otro valor, que será el *output*, a través de alguna función matemática. Por mencionar algunas de las más comunes en el ámbito de la redes neuronales artificales están la función sigmoidal, la función ReLU (unidad lineal rectificada) y la tangente hiperbólica. El papel que cumplen estas funciones es añadir no linealidad a las neuronas para que de esta forma se puedan concatenar diferentes capas.

Tóth (2015) dice que la función maxout, primero, fue propuesta por Goodfellow et al. (2013), y segundo, divide el número de neuronas N de cierta capa en L grupos de tamaño K ($N = K \cdot L$). Lo que hace esta función es aumentar la flexibilidad, ya que, por ejemplo, una función ReLU es una función definida a trozos lineal que está dividida en solo dos partes, mientras que la función de activación maxout está dividida en K partes. Gra-

cias a esta flexibilidad esta función ha demostrado mayor rendimiento en reconocimiento de imágenes (Goodfellow et al., 2013) y reconocimiento de fonemas (Tóth, 2015). Por ejemplo en el trabajo de Tóth (2015) se comparó el rendimiento de CNNs con función de activación maxout y ReLU en la prueba TIMIT y se encontró que la CNN con maxout consiguió mejores resultados que su contraparte con ReLU.

3.5. Técnica de entrenamiento dropout

Para solucionar el problema de *overfitting*, o al menos para disminuirlo un poco, Tóth (2015) utilizó la técnica de entrenamiento dropout, ya que, como mencionan Hinton et al. (2012) "Overfitting can be reduced by using 'dropout' to prevent complex co-adaptations on the training data.". La forma en que lo hace, la menciona Tóth (2015), en la cual el dropout omite cada una de las neuronas de una forma aleatoría con una probabilidad p. Investigaciones han mostrado que el dropout funciona óptimamente al momento de entrenar redes neuronales que trabajan con función de activación maxout (Miao et al., 2013; Swietojanski et al., 2014; Cai et al., 2014 en Tóth, 2015). Finalmente, en el trabajo de Tóth (2015) se observó un pequeño aumento en el rendimiento en las CNNs entrenadas con dropout, en este caso variando las tasas de dropout en cada capa en un tamaño de 0.25, que dio los mejores resultados en la optimización.

4. Redes Neuronales Recurrentes

Las redes neuronales recurrentes (RNNs por sus siglas en inglés) son un tipo de DNNs, que, al igual que las CNNs, tiene diversas aplicaciones en la ciencia aplicada, entre ellas el reconocimiento de escritura "a mano alzada", composición músical, y por supuesto, reconocimiento del habla. Estas, como menciona Géron (2017), permiten predecir el futuro, gracias a su arquitectura y a cómo trabajan. Además, menciona el mismo autor, estás DNNs se caracterizan por ser capaces ser creativas, lo que les da la habilidad de predecir, por ejemplo, cuál es la siguiente nota en una melodía y de esa forma componer un pieza musical.

Lo que diferencia a este tipo de NNs de otras, es que no se limitan a recibir un *input* y a partir de este pasar un *output* a la siguiente neurona, de modo que ese se convierte en un *input* y así sucesivamente; sino que las neuronas recurrentes envían el *output* no solo a la neurona siguiente, sino también a sí misma, de modo que se convierte en un *input* para la misma neurona, así que cada neurona recurrente recibiría dos tipos de *input*, el de la neurona anterior y el propio que viene del estado anterior. Esto se puede entender mejor si se visualiza en una imagen, para ello véase la figura 2. Por este comportamiento es que llaman a estas redes neuronales *backforward neural networks*. Esta forma que tiene la neurona para retroalimentarse es lo que permite que haya una cierta memoria, que le permite recordar estados pasados hasta una cierta cantidad de épocas (o *epochs* en inglés) atrás. Por lo que la red puede, de esta manera, aprender dinámicas temporales complejas en secuencias.

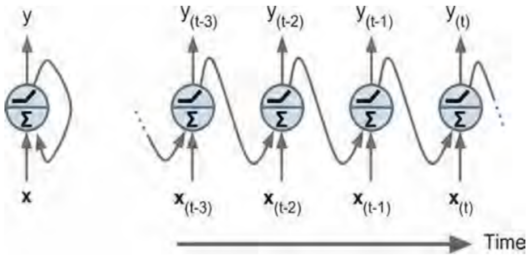

Figura 2: Neurona recurrente (izquierda) y neurona recurrente desenrrollada en el tiempo (derecha). Imagen tomada de Géron (2017)

Las RNNs han demostrado tener un rendimiento muy alto en lo que concierne al reconocimiento del habla, inclusive llegando a superar a las llamadas *feedforward neural networks* en tareas de reconocimiento del habla, como en Sak et al. (2015) y Miao et al. (2015); inclusive en tareas de reconocimiento de gran escala, como en los trabajos de Sak et al. (2014a) y Sak et al. (2014b). Además estos modelos han utilizado diferentes tipos de técnicas, como la clasificación temporal conexionista (CTC) o el uso de modelos de *long-short term memory* (LSTM), para mejorar el rendimiento y solucionar ciertos problemas que surgen al trabajar con estas NNs, algunos de los cuales serán ampliados en las siguientes subsecciones.

4.1. Problema del desvanecimiento del gradiente

Como ya se mencionó anteriormente, este tipo de NNs funcionan como una memoria que puede ir almacenando los datos de estados pasados de la misma red, el problema que surge con esto es que solo se puede trabajar hasta cierta cantidad de *epochs* en el pasado, por lo que si se depende de una "memoria de largo plazo"se puede complicar

todo debido al problema del desvanecimiento del gradiente.

Para entender de qué trata este problema, es necesario primero saber qué es el *backpropagation* y qué es el descenso del gradiente. Para plasmarlo de una manera resumida y que sea fácil de entender, el descenso del gradiente es un algoritmo de optimización utilizado a la hora de entrenar las NNs, que permite encontrar el mínimo de una función buscando así obtener el menor error posible. Todo se hace por medio de la realización de derivadas parciales de cada uno de los parámetros. Al hacer eso se obtiene un valor que debe ser "cambiado de dirección" ya que esto es a lo que se le conoce como el ascenso del gradiente, y lo que se quiere es lo contrario. Esto le permite saber al algoritmo cuál es el mínimo de la función y dónde se comete el menor error. La derivada parcial de la función de error se va pasando de una capa hacia otra hacia atrás hasta que se logre ajustar los pesos de las neuronas que más error producen, de allí que se llame *backpropagation* (o propagación hacia atrás, en español). Es decir, lo que hace el algoritmo de *backpropagation* es enviar el error hacia atrás para que las otras neuronas ejecuten el algoritmo de descenso del gradiente. El problema con esto es que, como lo que se propaga hacia atrás es "un poco del error", es decir esa derivada parcial, cada vez lo que se pasa a las capas anteriores es menor, de modo que no afecta significativamente a las neuronas que lo reciben a lo último de la red, haciendo que no se corrijan los pesos asignados a los *inputs* y no se mejore el error, llegando inclusive a detener el entrenamiento de toda la red.

Se han propuesto varias soluciones a este problema, pero acá se tratará solamente la *long-short term memory*, por ser la utilizada en tareas de reconocimiento del habla (Miao et al., 2015; Sak et al., 2015, 2014b,a)

4.1.1. Long short-term memory

Acá no se entrará en detallen en lo que es la LSTM, ya que esto requeriría de un trabajo tan extenso como puede ser este. Solamente se describirá la forma en que este soluciona el problema del desvanecimiento del gradiente.

Básicamente, la LSTM permite almacenar o memorizar valores en intervalos de tiempo aleatorios mientras que las tres puertas que componen las célula LSTM regulan el flujo de la información. La forma en que la LSTM soluciona el problema del desvanecimiento del gradiente es permitiendo que los gradientes fluyan hacia atrás sin sufrir ningún cambio, por lo que de esta forma el error llega a las capas más lejanas de la red, mejorando así su entrenamiento.

El uso de LSTM es tan común dentro de las investigaciones de reconocimiento del habla que ya se nombró a la combinación de esta con las RNN como LSTM RNN, más especificamente, se han utilizado LSTM RNN profundas, que son varias de estas agrupadas en una sola red.

4.1.2. Clasificación temporal conexionista

La CTC es una técnica utilizada para etiquetar secuencias en tiempos variables donde la alineación entre los *inputs* y las etiquetas objetivo es desconocida (Sak et al., 2015). En esta técnica las etiquetas "correctas", por llamarlas de alguna manera, son asignadas sobre otras debido a su mayor probabilidad de aparición. Se diferencian, además, de los modelos convencionales en dos cosas principalmente, las cuales las mencionan Sak et al. (2015): la existencia de una etiqueta en blanco, dado el caso que la red no perciba nada parecido a las etiquetas; y los criterios de entrenamiento que permiten la optimización de obtener la probabilidad para ciertas etiquetas. En el trabajo de Sak et al. (2015), por ejemplo, y otros más se utilizaron arquitecturas de LSTM RNN en combinación con un entrenamiento de CTC y han demostrado superar a las LSTM RNN convencionales en rendimiento (Sak et al., 2015; Miao et al., 2015).

4.2. Redes neuronales recurrentes bidireccionales profundas

Las RNN bidireccionales profundas son capaces de predecir elementos de una secuencia, al igual que una RNN nomal, solo que esta combina dos o más capas de manera paralela, unas analizando los *inputs* de izquierda a derecha y las otras de derecha a izquierda, o lo que es lo mismo hacia adelante y hacia atrás. Todo esto mejora el redimiento de la red neuronal y son bastante útiles al combinarlas con LSTM RNN. Por ejemplo en la investigación heca por Sak et al. (2015), se utilizó varias capas de LSTM.

Las RNN bidireccionales profundas han demostrado tener un mayor rendimiento que aquellos modelos de reconocimiento del habla que no las implantan (Graves et al., 2013,?; Sak et al., 2014a).

5. Discusión y conclusión

En este artículo se describió la forma en que se han utilizado las RNNs y CNNs en el reconocimiento del habla y como se utilizarían en la metodología que aquí se propone para la transcripción automática, donde el papel de las CNNs sería, como lo menciona Deng and Chen (2014), "to extract high-level acoustic features that are temporally much more smooth than the raw acoustic data sequence". Aunque lo anteriormente mencionado fue escrito para describir el papel de las DNNs, claramente es válido para una CNN. Todo esto quiere decir que las CNNs se deberían encargar de extraer características acústicas tanto en el eje de la frecuencia, como en el eje temporal, por razones que ya se mencionaron en la sección 3. Por otro lado, el papel de las RNNs, más específicamente LSTM RNNs con CTC, es el de clasificadoras de secuencias "performing the dynamic decoding task o derive linguistic sequences"(Deng and Chen, 2014).

De esta forma, con esta metodología mixta que utiliza las CNNs y RNNs como ya se mencionó en el párrafo anterior, se logrará diseñar una herramienta que sea capaz de realizar transcripciones automáticas con dos grandes ventajas, la primera es que se desenvuelve en una tarea que puede llegar a ser extenuante para un lingüista, permitiéndole concentrarse en otro tipo de tareas como puede ser el análisis fonológico; mientras que la segunda ventaja implica un mejor desempeño en la transcripción puesto que esta ya no está ligada íntimamente a la subjetividad del lingüista que transcribe, disminuyendo así en gran medida la posibilidad de error. El lingüista será, pues, el encargado de diseñar las redes neuronales, en caso de que sepa programar, y juzgar la calidad de las transcripciones creadas por el modelo. Aunque en ambos casos debe tener conocimientos en el campo de la fonética acústica.

El trabajo futuro debería centrarse en poner a prueba empíricamente esta metodología y comprobar su funcionamiento para así avanzar más en el desarrollo de una herramienta que, si llega a tener cerca del 100 % de precisión, podría llegar a remplazar a un transcriptor humano.

Referencias

Abdel-Hamid, O., Deng, L., and Yu, D. (2013). Exploring convolutional neural network structures and optimization techniques for speech recognition. In *Interspeech*, volume 2013, pages 1173–5.

Abdel-Hamid, O., Mohamed, A.-r., Jiang, H., and Penn, G. (2012). Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In *2012 IEEE international conference on Acoustics, speech and signal processing (ICASSP)*, pages 4277–4280. IEEE.

Cai, M., Shi, Y., and Liu, J. (2014). Stochastic pooling maxout networks for low-resource speech recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3266–3270. IEEE.

Deng, L., Abdel-Hamid, O., and Yu, D. (2013). A deep convolutional neural network using heterogeneous pooling for trading acoustic invariance with phonetic confusion. In *ICASSP*, pages 6669–6673.

Deng, L. and Chen, J. (2014). Sequence classification using the high-level features extracted from deep neural networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6844–6848. IEEE.

Géron, A. (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. .º'Reilly Media, Inc.".

Goodfellow, I. J., Warde-Farley, D., Mirza, M., Courville, A., and Bengio, Y. (2013). Maxout networks. *arXiv preprint arXiv:1302.4389*.

Graves, A., Mohamed, A.-r., and Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Huang, J.-T., Li, J., and Gong, Y. (2015). An analysis of convolutional neural networks for speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4989–4993. IEEE.

Khan, S., Rahmani, H., Shah, S. A. A., and Bennamoun, M. (2018). A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision*, 8(1):1–207.

Miao, Y., Gowayyed, M., and Metze, F. (2015). Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 167–174. IEEE.

Miao, Y., Metze, F., and Rawat, S. (2013). Deep maxout networks for low-resource speech recognition. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 398–403. IEEE.

Sainath, T. N., Kingsbury, B., Mohamed, A.-r., Dahl, G. E., Saon, G., Soltau, H., Beran, T., Aravkin, A. Y., and Ramabhadran, B. (2013a). Improvements to deep convolutional neural networks for lvcsr. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 315–320. IEEE.

Sainath, T. N., Kingsbury, B., Saon, G., Soltau, H., Mohamed, A.-r., Dahl, G., and Ramabhadran, B. (2015). Deep convolutional neural networks for large-scale speech tasks. *Neural Networks*, 64:39–48.

Sainath, T. N., Mohamed, A.-r., Kingsbury, B., and Ramabhadran, B. (2013b). Deep convolutional neural networks for lvcsr. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8614–8618. IEEE.

Sak, H., Senior, A., and Beaufays, F. (2014a). Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*.

Sak, H., Senior, A., Rao, K., and Beaufays, F. (2015). Fast and accurate recurrent neural network acoustic models for speech recognition. *arXiv preprint arXiv:1507.06947*.

Sak, H., Vinyals, O., Heigold, G., Senior, A., McDermott, E., Monga, R., and Mao, M. (2014b). Sequence discriminative distributed training of long short-term memory recurrent neural networks. In *Fifteenth annual conference of the international speech communication association*.

Swietojanski, P., Li, J., and Huang, J.-T. (2014). Investigation of maxout networks for speech recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7649–7653. IEEE.

Tóth, L. (2014). Combining time-and frequency-domain convolution in convolutional neural network-based phone recognition. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 190–194. IEEE.

Tóth, L. (2015). Phone recognition with hierarchical convolutional deep maxout networks. *EURASIP Journal on Audio, Speech, and Music Processing*, 2015(1):25.

Zhang, X., Trmal, J., Povey, D., and Khudanpur, S. (2014). Improving deep neural network acoustic models using generalized maxout networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 215–219. IEEE.