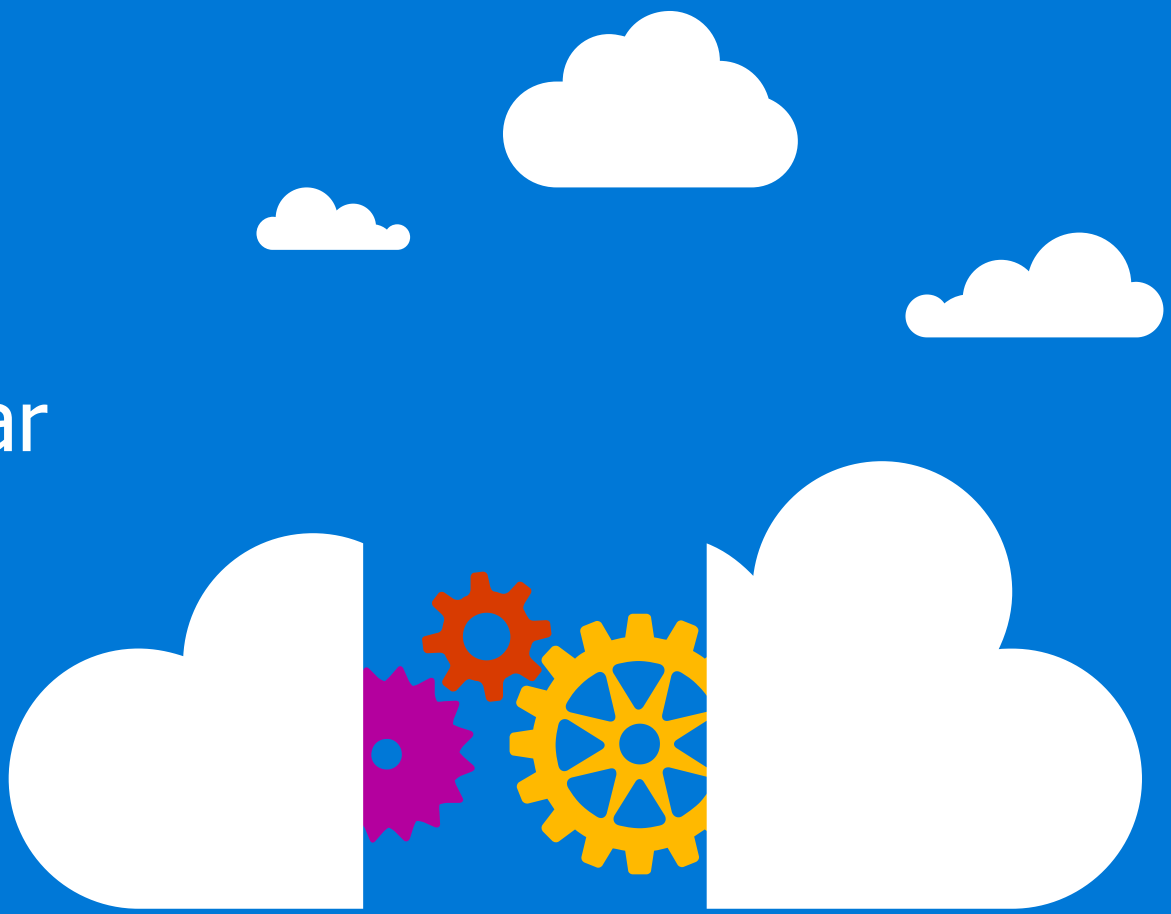
Curriculum Learning in Deep Neural Networks for Financial Forecasting

Allison Koenecke & Amita Gajewar
koenecke@stanford.edu
amitag@microsoft.com



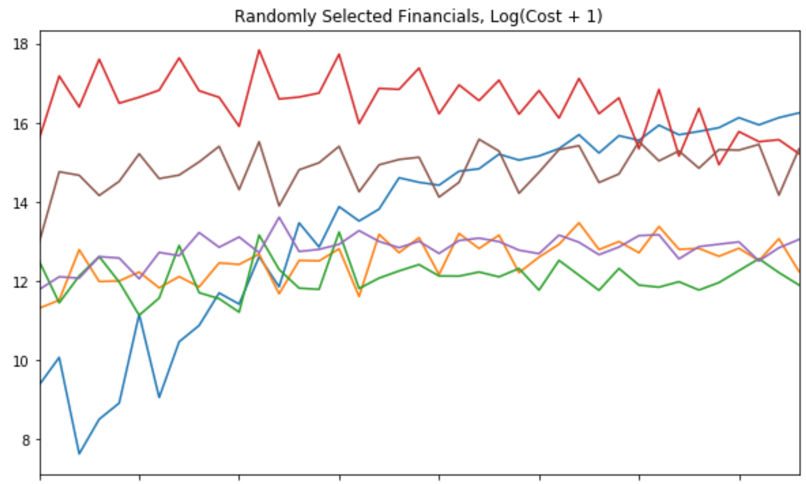
Presented at MIDAS 2019

Microsoft

Problem Statement

- **Goal**
  - Explore potential of using deep neural network (DNN)-based models to make more accurate predictions of Microsoft's revenue for each product
  - Compare the DNN-based model performance with previously developed ensemble models [1] combining traditional statistical methods (e.g., ETS, ARIMA) and basic ML models (e.g., random forest) deployed into production for comparison purpose

- **Data**
  - Time series of Microsoft revenue for each fiscal quarter from FY2009 Q1 to FY2018 Q3, for each product
    - 8 product segments
    - 60 regions / segment
    - 20 products / region



[1] Improving Regional Revenue Forecasts using Product Hierarchy, Amita Gajewar. *International Symposium on Forecasting 2018*.

Motivation

- **Prior work using deep learning methods**
  - Are mostly run on “big” data whereas our data are medium-sized
  - Mostly use basic stacked LSTMs rather than advanced techniques
  - Rarely compare multiple pre-processing mechanisms
  - Usually do not use categorical data as inputs alongside time series

- **In addition to covering the above, we present novel ideas in that**
  - Model techniques are borrowed from other applications (NLP and computer vision) and applied to time series forecasting
  - We invoke a novel application of curriculum learning as applied to time series

Data Pre-Processing

- **Data Cleaning**
  - Adjust all revenue values to USD with a constant exchange rage
  - Remove rows if over half of the timesteps have missing values, or if actual values within past 4 quarters are missing
- **Within-Task Transfer Learning**
  - Only train model on time series with at least 6 years of data
  - The 84% of datarows with enough historical data are trained and applied to datarows lacking enough data for inputs

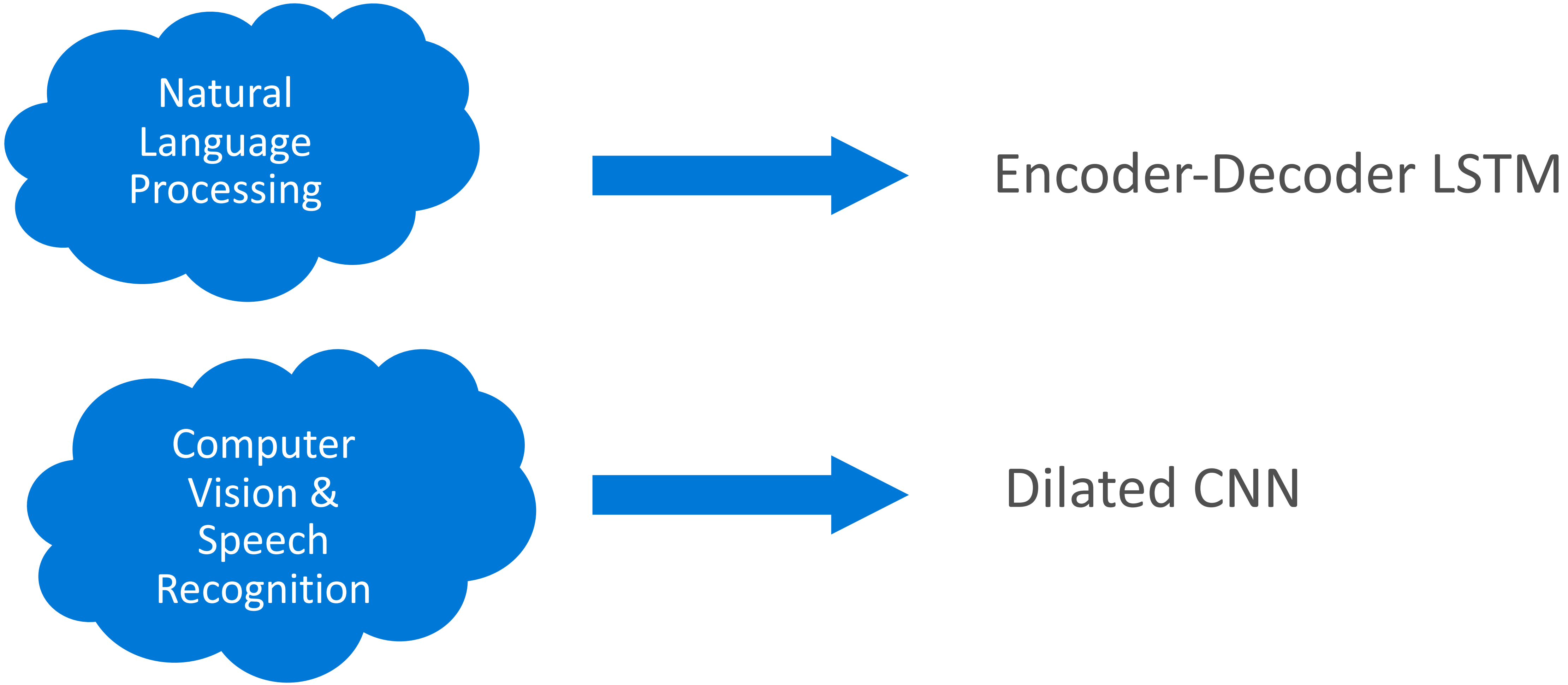
	Group	SubRegionName	CustomSubsegment	CustomSRSD	2009-01-01 00:00:00	2009-04-01 00:00:00	2009-07-01 00:00:00	2009-10-01 00:00:00	2010-01-01 00:00:00
0	Argentina . Commercial Enterprise . Developer ...	Argentina	Commercial Enterprise	Developer Tools					
1	Argentina . Commercial Enterprise . Dynamics O...	Argentina	Commercial Enterprise	Dynamics OnPrem					

$$$

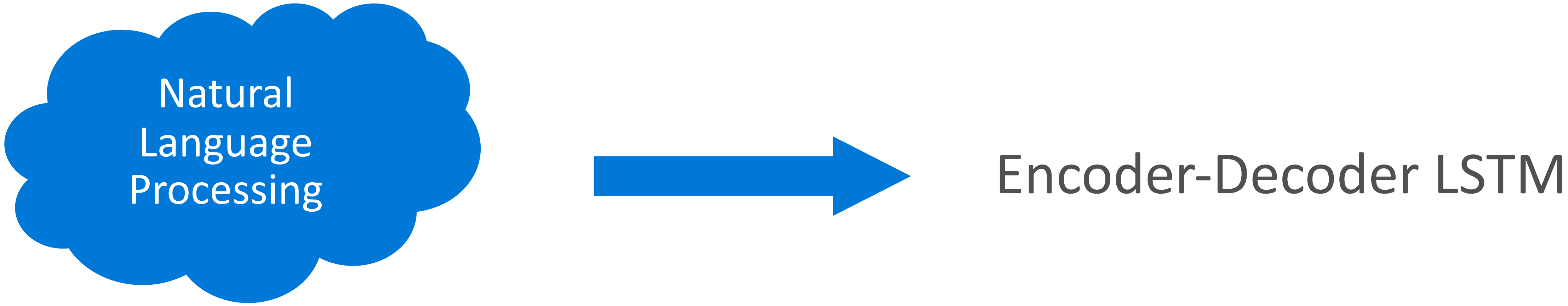
~ 6,000 datarows

39 timesteps

Two DNN Methods



Model #1



Variants of Encoder-Decoder LSTM

1. Basic Encoder-Decoder LSTM
2. LSTM with Categorical Indicators
3. LSTM with Seasonality
4. LSTM with Curriculum Learning

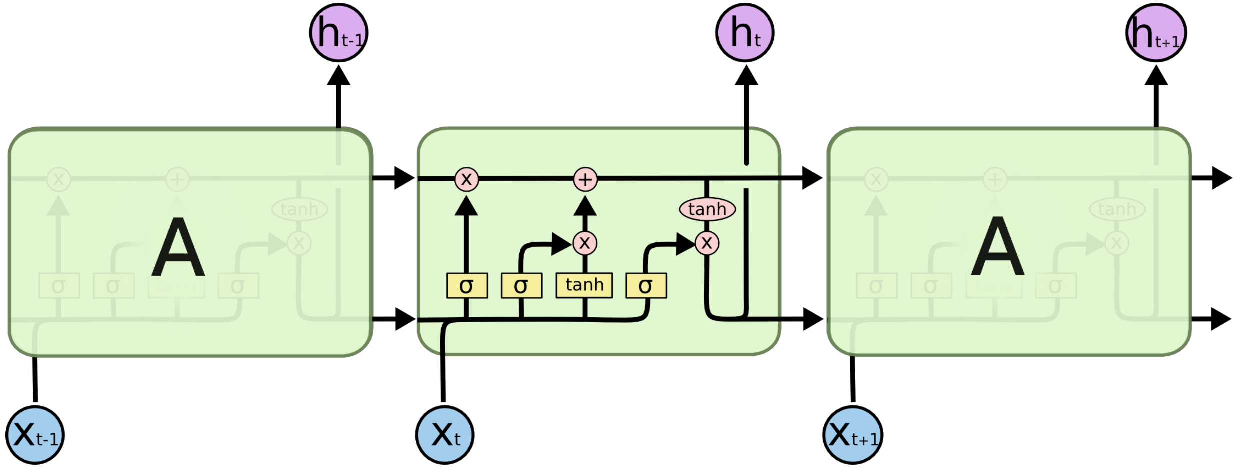


Image source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM Pre-Processing

Smoothing Transformation

- Calculate log(revenue+1) and then de-mean data within each of training and validation sets

Walk-Forward Split

- Step forward into time for training vs. validation to ensure no data leakage
- Do this iteratively for several 15-time-step-sized windows within the data

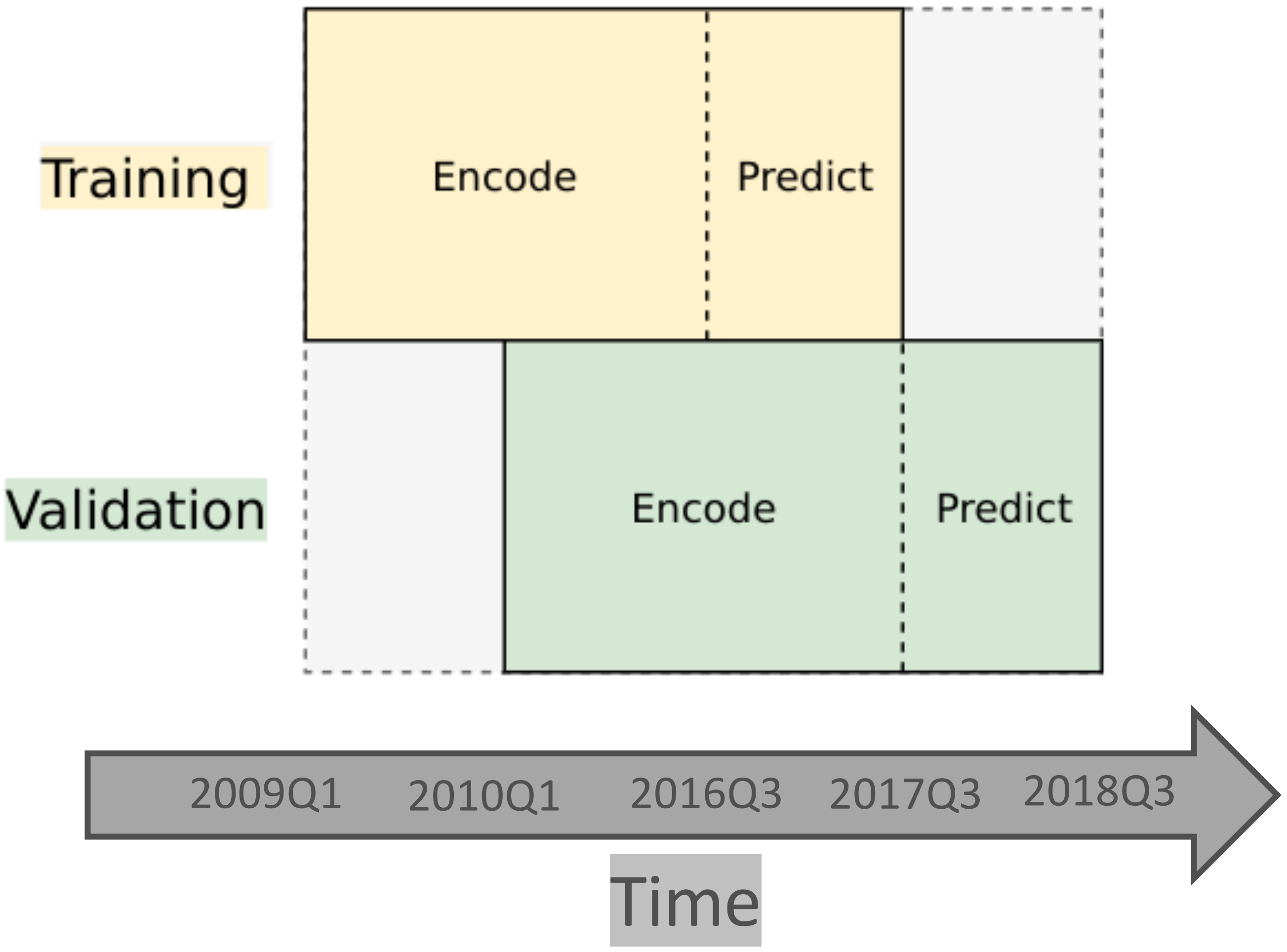


Image source: https://github.com/Arturus/kaggle-web-traffic/blob/master/how_it_works.md

LSTM Architecture

- **Seq2Seq model**
  - LSTM encoder: process revenue and return internal state
  - LSTM decoder: use previous time step's actual data and internal LSTM encoder states to generate next output
  - Adam optimizer on MAE
  - Train for each rolling window, e.g. fit model at first 15 time steps; step forward 4 steps and repeat
- **Inference**
  - Teacher forcing during training (feed predicted rather than true value as next input)
  - Decode and inverse smoothing transformation for the last 4 quarters of data

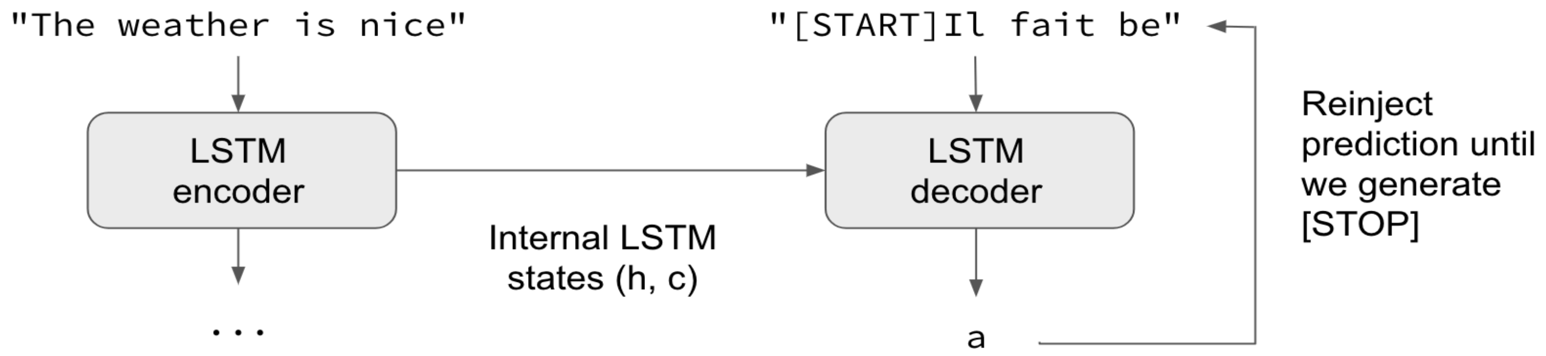
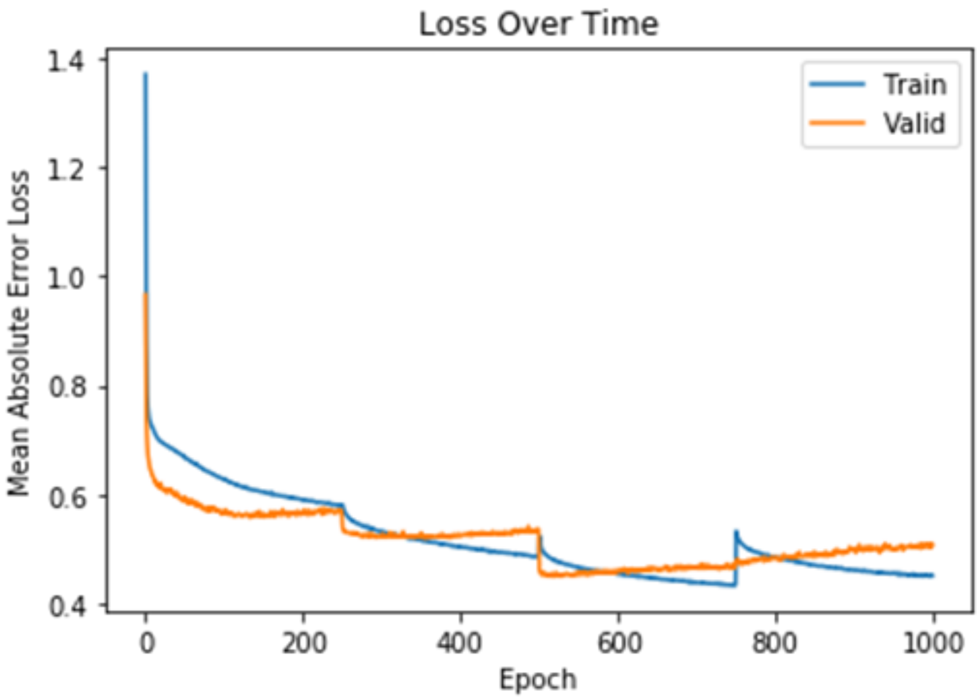


Image source: https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html

Basic LSTM: Example Loss Curve

- Training and validation loss curves as model is fit iteratively through rolling windows of predefined step size
- Notice gradual loss as model uses previous weights to warm-start rather than fitting from the scratch
- Significantly improved from vanilla LSTM model, which overfits



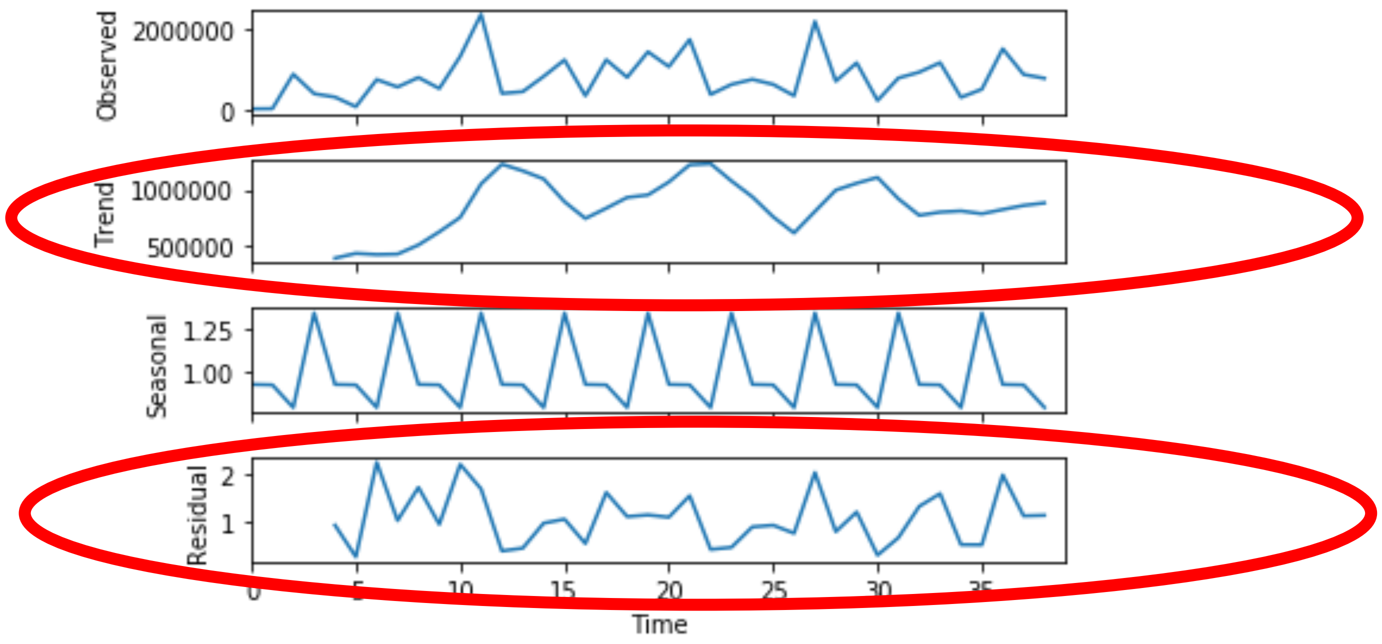
LSTM with Categorical Variables

	Group	SubRegionName	CustomSubsegment	CustomSRSD	2009-01-01 00:00:00	2009-04-01 00:00:00	2009-07-01 00:00:00	2009-10-01 00:00:00	2010-01-01 00:00:00
0	Argentina . Commercial Enterprise . Developer ...	Argentina	Commercial Enterprise	Developer Tools					
1	Argentina . Commercial Enterprise . Dynamics O...	Argentina	Commercial Enterprise	Dynamics OnPrem					

One-Hot Vectors

LSTM with Seasonality

- **Seasonal Decomposition**
  - Use multiplicative seasonal decomposition (more revenue → more seasonality observed)
  - Requires all values to be strictly positive; hence, 0 values are substituted as ones
  - Resulting components: trend, seasonal, and residual
- **Pre-processing**
  - Calculate trend*residual values in training and validation sets; use trend*residual as training input
  - Smoothing Transformation: take log(revenue+1) and de-mean using de-seasonalized training data values



LSTM with Curriculum Learning

- Cleverly changing the order of inputs to a model can improve results
- Example: NLP
    - Intuition: shorter sentences are easier to learn than longer sentences
    - Bootstrapping via iterated learning of increasingly longer sentences; requires no initialization
- Baby Steps algorithm on model M, training data D, and curriculum C

Algorithm 2 Baby Steps Curriculum

procedure BS-CURRICULUM(M,$\mathcal{D}$, $\mathcal{C}$)
    $\mathcal{D}' = \text{sort}(\mathcal{D}, \mathcal{C})$
    $\{\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^k\} = \mathcal{D}'$ where $\mathcal{C}(d_a) < \mathcal{C}(d_b)$ $d_a \in D^i$, $d_b \in D^j$, $\forall i < j$
    $\mathcal{D}^{train} = \emptyset$
    for $s = 1...k$ **do**
        $\mathcal{D}^{train} = \mathcal{D}^{train} \cup \mathcal{D}^s$
        while not converged for p epochs **do**
            train(M, $\mathcal{D}^{train}$)
        end while
    end for
end procedure

Image source: https://arxiv.org/pdf/1611.06204.pdf

LSTM with Curriculum Learning

- What curriculum to use?
  - Sort metric: residual trend (from STL) weighted by segment revenue for each datarow
  - Sort order: train in increasing order of residual
  - Create *k* batches from the sorted datarows, in sort order
  - Within each batch, shuffle the datarows during training

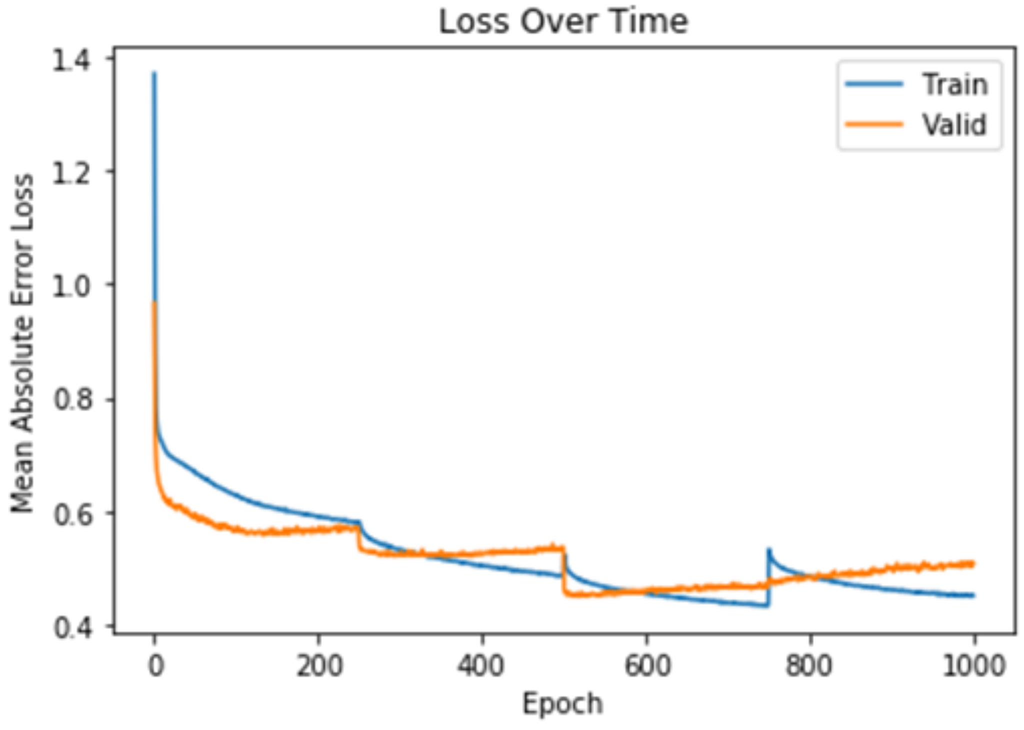
for $s = 1 \ldots k$ **do**
  $\mathcal{D}^{train} = \mathcal{D}^{train} \cup \mathcal{D}^{s}$
  while not converged for p epochs **do**
    train$(M, \mathcal{D}^{train})$

- Within each iteration of the rolling window process
  - Continue warm-start by iteratively adding one batch at a time to the training data
  - Each rolling window iteration is run p times, where p is the # epochs chosen to reach convergence

Image source: https://arxiv.org/pdf/1611.06204.pdf

LSTM Example Loss Curves

Only Rolling Windows



Rolling Windows + Curriculum Learning

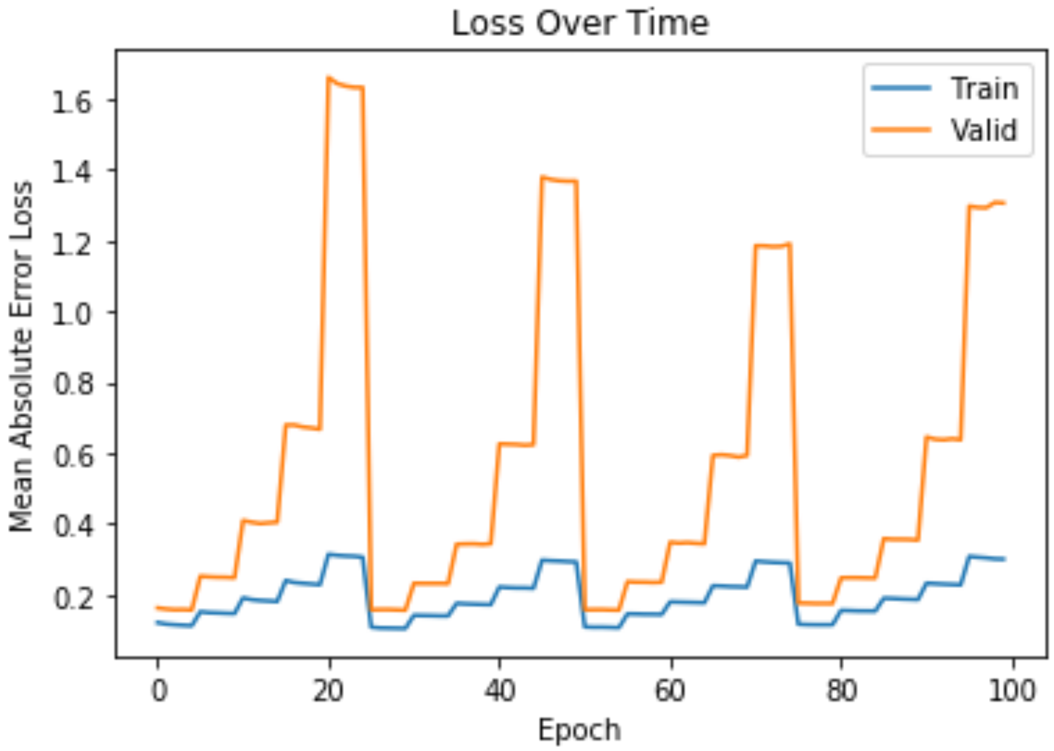
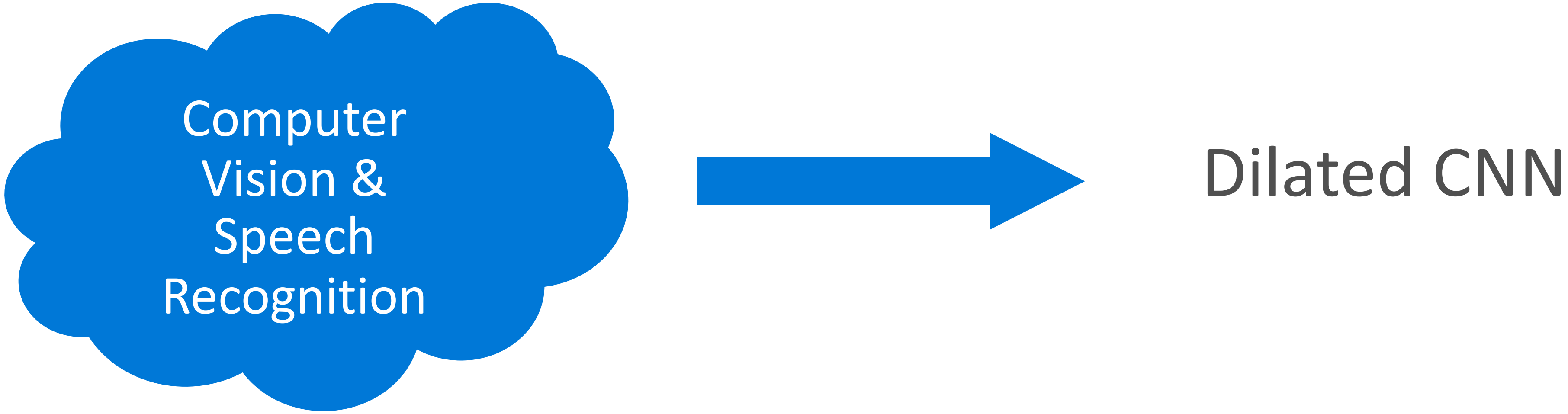


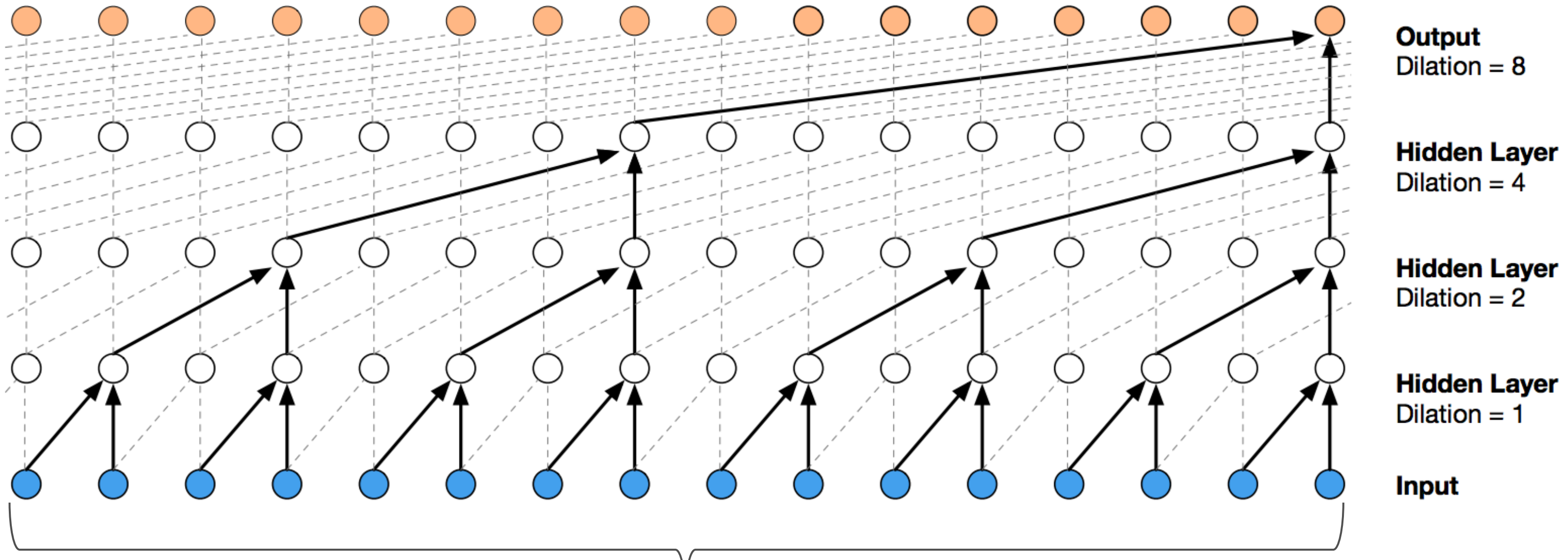
Image source: https://arxiv.org/pdf/1611.06204.pdf

Model #2



Variants of Dilated CNN

1. Basic Dilated CNN
2. Dilated CNN with Categorical Indicators
3. Dilated CNN with Curriculum Learning



No more need for rolling windows or seasonality given long history as input

Image source: https://deepmind.com/blog/wavenet-generative-model-raw-audio/

Basic Dilated CNN

- **Smoothing Transformation**
  - Take log(revenue+1) and de-mean using the training data values
- **Dilated CNN Model**
  - Use 1D Convolutions in each of 10 dilated convolutional layers:

    6 filters of width 2 per layer; using a small number of dilated causal convolutional layers can connect an exponential number (2^10) of input values for the output
  - Use two fully connected layers to get final output: Dense(128) with ReLU activation; Dense(1)
  - Adam optimizer on MAE
- **Inference**
  - Teacher forcing during training; predict the last 4 quarters of data iteratively; append each to history

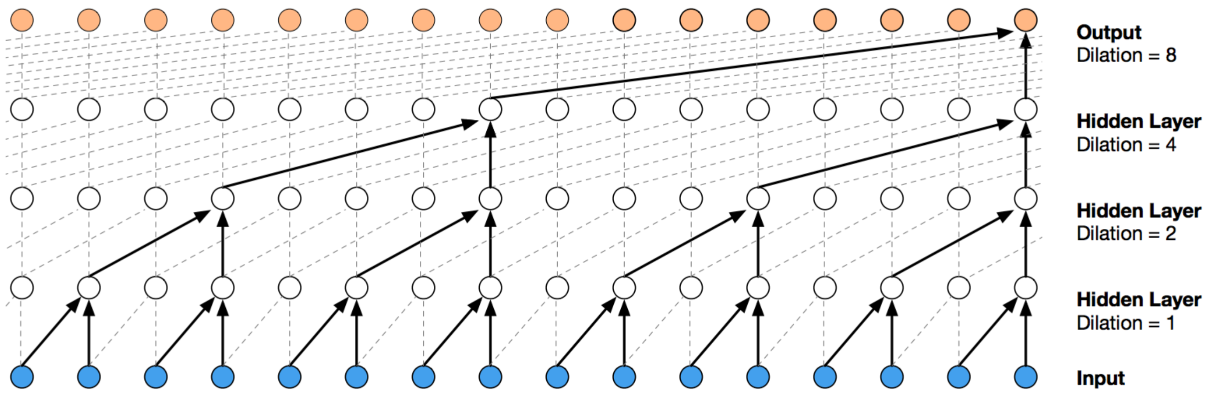
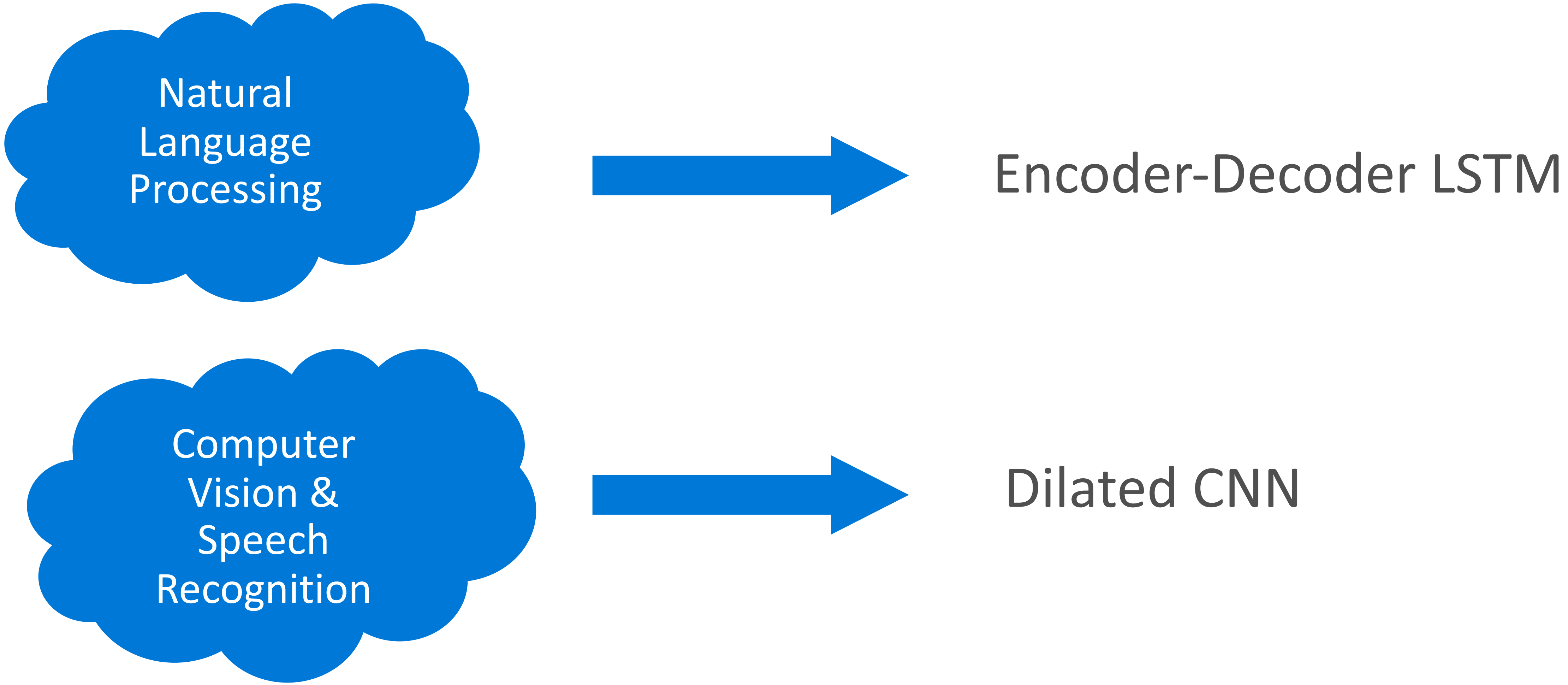


Image source: https://deepmind.com/blog/wavenet-generative-model-raw-audio/

Results


Natural Language Processing
Encoder-Decoder LSTM
Computer Vision & Speech Recognition
Dilated CNN

Model Evaluation Metrics

- **Test data**
  - Models were evaluated using the last four quarters' data as the test set (FY17Q4 to 18Q3)
- **Randomness introduced by DNN models**
  - We run each experiment 30 times and ensemble the results to reduce inherent variance
- **Error evaluation**
  - MAPE is calculated by considering the average of the forecasts across runs as the final forecast, and comparing that to actual observed revenue for the corresponding quarter
  - Worldwide forecast = sum of forecasts for all rows
  - Segment forecast = sum of all SRSD forecasts that fall into the segment
  - Compare Worldwide and Segment-level MAPEs to Microsoft baseline MAPEs (actual MAPEs excluded for privacy reasons)

LSTM + Curriculum Learning Improves Accuracy

Table 1. World-wide test error reduction percentages of DNN models over previous Microsoft production baseline.

Model	**Percent MAPE Improvement**
Basic LSTM	1.9%
LSTM with Categorical Indicators	18.2%
LSTM with Seasonality	-5.1%
LSTM with Curriculum Learning	**27.0%**
Basic DCNN	-0.7%
DCNN with Categorical Indicators	12.1%
DCNN with Curriculum Learning	**22.6%**

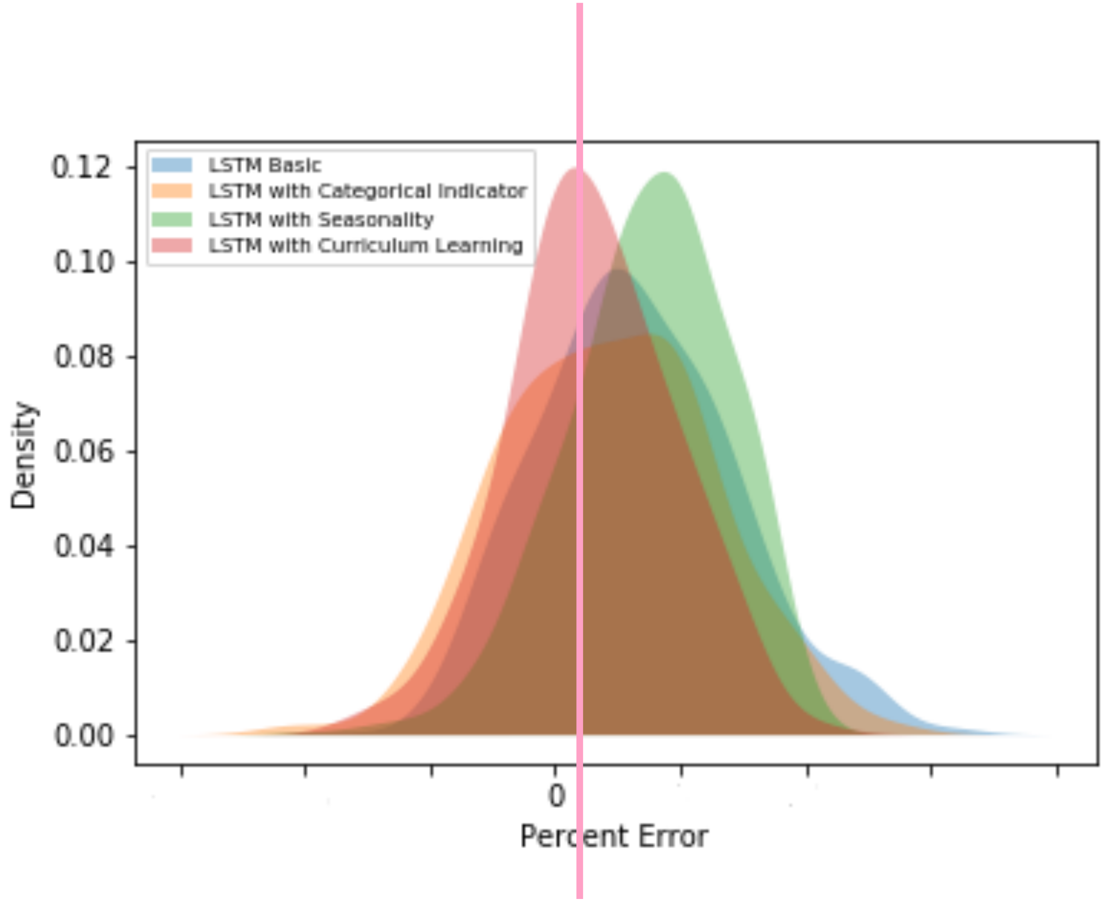
Table 2. LSTM Model Segment-level MAPE reduction percentages (%) over previous Microsoft production baseline (positive % corresponds to error reduction).

Segment	**Basic LSTM** *(Model (a))*	**Model (a) + Categorical Indicators** *(Model (b))*	**Model(b) + Seasonality** *(Model (c))*	**Model(c) + Curriculum Learning** *(Model(d))*
1	25.5	22.0	53.4	70.0
2	-47.9	-34.3	-23.0	-0.8
3	7.65	-5.8	26.0	20.3
4	14.2	30.3	12.0	27.4
5	-15.4	-13.2	-11.8	-25.9
6	-79.2	-60.3	-110.1	-12.4
7	34.7	30.1	31.0	11.5
8	17.9	15.5	57.2	61.4
Revenue-weighted Average	**10.3**	**10.3**	**21.3**	**30.0**

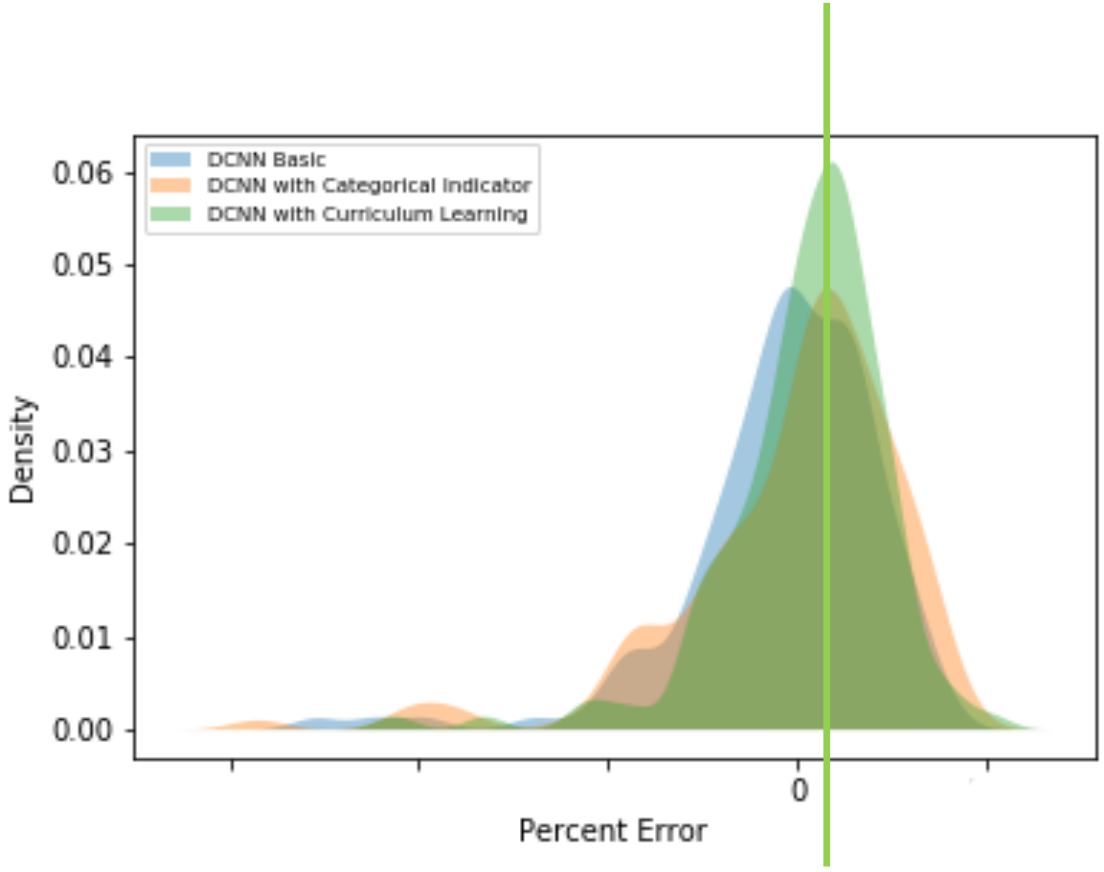
Table 3. DCNN Model Segment-level MAPE reduction percentages (%) over previous Microsoft production baseline (positive % corresponds to error reduction).

Segment	**Basic DCNN** *(Model (a))*	**Model (a) + Categorical Indicators** *(Model (b))*	**Model(b) + Curriculum Learning** *(Model (c))*
1	24.8	44.0	34.2
2	-0.2	-19.5	-19.5
3	-8.7	28.9	39.9
4	35.5	35.4	22.6
5	45.4	58.4	26.8
6	-258.2	-263.2	-80.5
7	27.0	28.7	29.4
8	33.8	35.5	24.9
Revenue-weighted Average	**-3.1**	**4.5**	**16.2**

Worldwide MAPE Densities: Low Bias & Variance



LSTM with Curriculum Learning



Dilated CNN with Curriculum Learning

Conclusions

- **Curriculum learning is a powerful technique to explore**
  - Applying a good sorting metric to NN inputs can improve results drastically
  - In practice, run times are significantly faster, especially on medium-sized data

- **Encoder-decoder LSTMs and Dilated CNNs can applied to time series data**
  - Models are fast and accurate; do not overfit

- **Seasonal decomposition can be a useful pre-processing tool**
  - Especially for financial data with steady seasonality

Future Work

- **Curriculum learning**
  - Further experimentation of different sorting orders
  - Ensembling across different orders to yield better segment-level MAPEs
  - Incorporate batch metrics into hyperparameter-tuning packages
- **Categorical variables for segmentation**
  - Try changing sample weights
  - Exploit hierarchical nature of categories better
- **Dilated LSTMs**
  - Would serve as an in-between of encoder/decoder LSTMs and dilated CNNs

Thank you.
Questions?