# Papers ArXiv 📔

`Summaries of the papers I read` ✏️

These notes are best viewed with MathJax extension in chrome.

`Dec 2, 2018`

**Wavenet: Generative model for raw audio**

- Link, post
- Raw audio is difficult to model because it has like 16000 samples per second. (And Completely auto-regressive).
- PixelRNN and PixelCNN generate images several thousand pixels at a time, inspired the 2-D Nets to a 1-D WaveNet!
- The same concept of products of conditional probabilities of previous timesteps is used by wavenet: $p(x) = \prod_{t=1}^{T} p(x_t | x_1, .., x_{t-1})$

- A stack of convolutional layers is used. Output is fed back to the network to generate next sample. Causal convolutions are used - which means the output depends on only previous items in the layer before it (But, the network needs to be very deep to increase receptive field)
- Dilated convolutions are used (Output the same size as input) Dilation Factor 1 = normal convolution. Helps networks to have a huge receptive field with just a few layers. This model also uses residual links and skip connections to train deep models.
- It uses softmax to discretize the outputs. But per each timestep, it will need to output $2^{16}$ probabilities. For this, it uses the mu-law to create 256 possibilities from it instead (called quantization).
  $f(x_t) = sign(x_t)\frac{ln(1+\mu|x_t|)}{ln(1+\mu)}$ where $-1 < x_t < 1$ and $\mu = 255$
- It also uses gated-activation instead of RELUs:
  $z = tanh(W * x) \odot \sigma(W * x)$ (Here $\odot$ = elementwise product and * = convolution)
- Conditional WaveNets: get p(x|h) just like in PixelCNNs. (In this case, the h would be speaker's embeddings in a text-to-speech model (Exactly a one-hot vector containing speaker-ID)) This would help generate speech samples from the speaker:
  $z = tanh(W * x + V^T h) \odot \sigma(W * x + V^T h)$ (Look at the Conditional Gated PixelCNN section below).
- Model consists of several blocks, each contining a bunch of layers with increasing dialating factors. (AS we go deeper, the receptive field increases).

`Nov 30, 2018`

### Conditional Image generation with PixelCNN Decoders

- Link
- Aims at conditional image modelling, as an image is generated pixel by pixel so the joint image distribution will be a product of conditionals (probability that a pixel will occur given the existing pixels (Autoregressive!))
- Usually PixelRNN performs better than this (Because all the previous pixel information is available to all layers of LSTMs but only to the last layers of the convolutional stack (as the receptive field grows as we go deeper), also

because LSTMs use gates(sigmoid and tanh of dot products) which can model more complex interactions instead of the usual RELUs), but this is much faster to train!

- **Cö̭ǹṯřibùṯiö̭ǹš**:
    - **Gated PixelCNN**: Image can be described as a product of conditional distributions of pixels: $p(x) = \prod_{i=1}^{n^2} p(x_i|x_1, .. x_{i-1})$. Each of these conditional distributions are modelled by a CNN in this technique!
        - This produces an output for an image of dimensions: (N x N x 3) as (N x N x 3 x 256) probabilities. And this can be produced parallely because of convolutions.
        - Gated convolutional layers: Replace RELUs with elementwise product of tanh and sigmoid of Wx- (To model more complex behaviours)
          $y = tanh(W * x) \odot \sigma(W * x)$ where $\odot$ is elementwise product and * is convolution.
        - RECAP Activations:
            - Sigmoid: (0 to 1): $f(x) = \frac{1}{1+e^{-x}}$ and $f'(x) = \frac{f(x)}{1-f(x)}$
            - TanH: (-1 to 1): $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ and $f'(x) = 1 - f(x)^2$
            - RELU: (0 to x): $f(x) = max(0, x)$ and $f'(x) = 1$ if x>0 else 0.
        - Uses a vertical and horizontal stack (Like PixelRN used the Diagonal BiLSTM) for having all pixels that occured before in the receptive field.
    - **Conditional Gated PixelCNN**:
        - Given a high level image description, given as a latent vector h (like a one-hot encoding of classes of ImageNet), it models the product of conditionals.
          $p(x|h) = \prod_{i=1}^{n^2} p(x_i|x_1, .. x_{i-1}, h)$ we do this by
          $y = tanh(W * x + V^T h) \odot \sigma(W * x + V^T h)$ to model if object is in the image (not where) Spatial coordinates can also be mapped if we use (V * s) in the above formula instead of $V^T h$ were s = spatialMapping(h).
        - Since it models the image distribution well based on latent vector representations, it will do well as a decoder in AutoEncoders!
        - This is also tried on face dataset with embeddings( h vector ) generated using FaceNet's(link) triplet loss function.

Nov 25, 2018

## Pixel Recurrent Neural Networks

- Link
- The aim of this paper is to estimate the distribution of images to compute the likelihood and generate new ones.
- Important obstacle in Generative Modeling is building something that is `tractable` (easy to understand) and `scalable`.
- **P̌řev̌̃iöùš ẁöřḱš**:
    - Techniques like AutoEncoders make use of Latent Variables (for dimensioanlity reduction) but are not tractable.
    - Tractable models involve modeling using products of conditional distributions, but are not sophisticated enough to model long-range correlation between pixels.
    - Enter, RNNs! They (2d-RNNs) have been awesome at modelling gray-scale images and textures before this paper.
        - Generating image pixel by pixel (sequential from top left) can be written as a product of conditional distributions: $p(x) = \prod_{i=1}^{n^2} p(x_i|x_1, ..x_{i-1})$
- **Cöǹṱřibùṱiöǹš**:
    - Row LSTM - Triangular Receptive Field
    - Diagonal BiLSTM - All pixels on left and top are in the receptive field.
    - Masked Convolutions - Values from R shouldnt be available when predicting G and B. To model full dependencies between color channels.
    - PixelCNN - Preserves spatial resolution (No pooling layers)
    - Using Softmax Discrete for pixel values 0-255 instead of continuous
    - Using Skip-connections helps with increased depth (12 layers).
    - Larger models = better results