Automatically Determining Hyperparameters
Grid Search v.s. Random Search

Assumption: top K results are good enough

If there are N points, probability K/N that your sample is in top K

Sample x times: \[1 - (1 - K/N)^x > 90\%

If N = 1000, K = 10 \[x = 230\]

If K = 100 \[x = 22\]
Model-based Hyperparameter Optimization

Reinforcement Learning

Design a network

Train the network

Accuracy as reward

One kind of meta learning (or learn to learn)

A Full Convolutional Neural Network (LeNet)
Learning Rate

- **Operands**: $g$, $g^2$, $g^3$, $\hat{m}$, $\hat{v}$, $\gamma$, sign($g$), sign($\hat{m}$), 1, 2, $\epsilon \sim N(0, 0.01)$, $10^{-4}w$, $10^{-3}w$, $10^{-2}w$, $10^{-1}w$, Adam and RMSProp.

- **Unary functions** which map input $x$ to: $x$, $-x$, $e^x$, $\log|x|$, $\sqrt{|x|}$, $\text{clip}(x, 10^{-5})$, $\text{clip}(x, 10^{-4})$, $\text{clip}(x, 10^{-3})$, $\text{drop}(x, 0.1)$, $\text{drop}(x, 0.3)$, $\text{drop}(x, 0.5)$ and sign($x$).

- **Binary functions** which map $(x, y)$ to $x + y$ (addition), $x - y$ (subtraction), $x \ast y$ (multiplication), $\frac{x}{y + \delta}$ (division), $x^y$ (exponentiation) or $x$ (keep left).
Can transfer to new tasks

\[ e^{\text{sign}(g) \times \text{sign}(m)} \times g \]
Activation Function

- **Unary functions:** $x, -x, |x|, x^2, x^3, \sqrt{x}, \beta x, x + \beta, \log(|x| + \epsilon), \exp(x) \sin(x), \cos(x), \sinh(x), \cosh(x), \tanh(x), \sinh^{-1}(x), \tan^{-1}(x), \text{sinc}(x), \max(x, 0), \min(x, 0), \sigma(x), \log(1 + \exp(x)), \exp(-x^2), \text{erf}(x), \beta$

- **Binary functions:** $x_1 + x_2, x_1 \cdot x_2, x_1 - x_2, \frac{x_1}{x_2 + \epsilon}, \max(x_1, x_2), \min(x_1, x_2), \sigma(x_1) \cdot x_2, \exp(-\beta(x_1 - x_2)^2), \exp(-\beta|x_1 - x_2|), \beta x_1 + (1 - \beta)x_2$
Activation Function

Graphs showing various activation functions:

1. $x \cdot \sigma(\beta x)$
2. $x \cdot (\sinh^{-1}(x))^2$
3. $\min(x, \sin(x))$
4. $(\tan^{-1}(x))^2 - x$

Graphs with other functions:

1. $\max(x, \sigma(x))$
2. $\cos(x) - x$
3. $\max(x, \tanh(x))$
4. $\text{sinc}(x) + x$
\[ a = z \cdot \text{sigmoid}(\beta z) \]
Neural Architecture Search with Reinforcement Learning

**LSTM**

From Reinforcement Learning
Computation Issue?

- Original version: 450 GPUs for 3-4 days (32,400-43,200 GPU hours).
- New version: Nvidia GTX 1080Ti GPU takes less than 16 hours.
- Main idea: forcing all child models to share weights to instead of training from scratch.

Hieu Pham, Melody Y. Guan, Barret Zoph, Quoc V. Le, Jeff Dean, “Efficient Neural Architecture Search via Parameter Sharing”, arXiv, 2018