Searching for activation functions

By Ang Ming Liang
About Me

I am an AI enthusiast

Interested in solving general intelligence

Will enter NUS Computational Biology program in 2019
Motivation

I am calling BS
AutoML

CLOUD AUTOML ALPHA
Train high quality custom machine learning models with minimum effort and machine learning expertise

Train Custom Machine Learning Models
Cloud AutoML is a suite of Machine Learning products that enables developers with limited machine learning expertise to train high quality models by leveraging Google’s state of the art transfer learning, and Neural Architecture Search Technology.

AutoML Vision is the first product to be released. It is a simple, secure and flexible ML service that lets you train custom vision models for your own use cases. Soon, Cloud AutoML will release other services for all other major fields of AI.
Quoc Le papers
And More !  #OPmuch
The idea

Sample architecture A with probability $p$

The controller (RNN)

Trains a child network with architecture A to get accuracy $R$

Compute gradient of $p$ and scale it by $R$ to update the controller
RNN part
Sampling functions

Core unit

\[ b(u_1(x_1), u_2(x_2)). \]
Sampling functions

- **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:** \( x_1 + x_2, x_1 \cdot x_2, x_1 - x_2, \frac{x_1}{x_2 + \epsilon}, \max(x_1, x_2), \min(x_1, x_2), \sigma(x_1) \cdot x_2, \exp(-\beta(x_1 - x_2)^2), \exp(-\beta|x_1 - x_2|), \beta x_1 + (1 - \beta)x_2 \)
Policy Gradients

Update the policy directly to maximise the expected long term rewards!

\[ \hat{g} = \hat{E}_t \left[ \nabla_\theta \log \pi_\theta(a_t \mid s_t) \hat{A}_t \right] \]

\[ A_t = \delta_t + (\gamma \lambda) \delta_{t+1} + \cdots + (\gamma \lambda)^{T-t+1} \delta_{T-1}, \]
where \( \delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \)

Leads to destructively large policy update!
Trust Region Methods

\[
\begin{align*}
\text{maximize} \quad & \hat{E}_t \left[ \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)} \hat{A}_t \right] \\
\text{subject to} \quad & \mathbb{E}_t [KL(\pi_{\theta_{\text{old}}} (\cdot | s_t), \pi_\theta (\cdot | s_t))] \leq \delta.
\end{align*}
\]

Surrogate function

Constraint based on the size of the policy update
Clipped Surrogate

\[
\hat{E}_t \left[ \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)} \hat{A}_t \right] = \hat{E}_t \left[ r_t(\theta) \hat{A}_t \right].
\]

\[
\text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t
\]

\[
L_{\text{CLIP}}(\theta) = \hat{E}_t \left[ \min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)\hat{A}_t) \right]
\]
Putting everything together

\[
L_t^{CLIP+VF+S}(\theta) = \hat{E}_t [L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)]
\]

\[
(V_\theta(s_t) - V_t^{\text{targ}})^2
\]

Entropy Bonus to increase exploration
Algorithm 1 PPO, Actor-Critic Style

for iteration=1,2,... do
    for actor=1,2,...,N do
        Run policy $\pi_{\theta_{\text{old}}}$ in environment for $T$ timesteps
        Compute advantage estimates $\hat{A}_1, \ldots, \hat{A}_T$
    end for
    Optimize surrogate $L$ wrt $\theta$, with $K$ epochs and minibatch size $M \leq NT$
    $\theta_{\text{old}} \leftarrow \theta$
end for
Results

INFO:tensorflow:global_step/sec: 74.7513
INFO:tensorflow:global_step/sec: 74.5161
INFO:tensorflow:loss = 2.307766, step = 15980378 (1.337 sec)
INFO:tensorflow:global_step/sec: 74.6956
INFO:tensorflow:loss = 2.2983905, step = 15980478 (1.342 sec)
INFO:tensorflow:global_step/sec: 74.7883
INFO:tensorflow:loss = 2.3078947, step = 15980578 (1.339 sec)
INFO:tensorflow:global_step/sec: 74.7883
INFO:tensorflow:loss = 2.3001652, step = 15980678 (1.337 sec)
INFO:tensorflow:global_step/sec: 74.8899
INFO:tensorflow:loss = 2.3074267, step = 15980778 (1.337 sec)
INFO:tensorflow:global_step/sec: 76.1826
INFO:tensorflow:loss = 2.3065348, step = 15980878 (1.313 sec)
INFO:tensorflow:global_step/sec: 75.3713
INFO:tensorflow:loss = 2.3012478, step = 15980978 (1.327 sec)
INFO:tensorflow:global_step/sec: 75.647
INFO:tensorflow:loss = 2.3102431, step = 15981078 (1.322 sec)
INFO:tensorflow:global_step/sec: 75.9671
INFO:tensorflow:loss = 2.318913, step = 15981178 (1.316 sec)
INFO:tensorflow:global_step/sec: 73.7092
INFO:tensorflow:loss = 2.3001149, step = 15981278 (1.357 sec)
INFO:tensorflow:global_step/sec: 74.0312
INFO:tensorflow:loss = 2.3001056, step = 15981378 (1.350 sec)
INFO:tensorflow:global_step/sec: 74.6670
INFO:tensorflow:loss = 2.3046515, step = 15981478 (1.340 sec)

**Activation functions**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3x</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>-3</td>
<td></td>
</tr>
</tbody>
</table>

Negative result :(
Why did we fail?

1. Didn’t train the model enough

<table>
<thead>
<tr>
<th>What if You</th>
<th>Wanted to reproduce a paper</th>
</tr>
</thead>
<tbody>
<tr>
<td>But it said</td>
<td>We've trained our model on 500 GPUs for 2 weeks</td>
</tr>
</tbody>
</table>
Why did we fail?

2. Local minima and saddle points
Swish

\[ x \cdot \text{sigmoid}(x) \]
Training using swish

Using SGD
Training using swish

Using RMSprop
Visualisation

RELU

SWISH

TANH
The future ?
Hyperparameter search
Github link

bit.ly/saf_git
Thank you

Email: angmingliang4017ic@gmail.com
Facebook: Ang Ming Liang