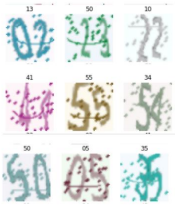
Captcha Solving

By Isaiah Brandt-Sims
&
Thunder Keck

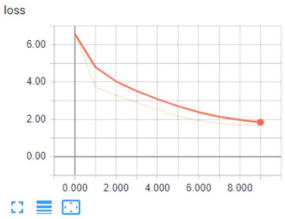
Predicting

Our model attempts to determine the string encoded in a captcha. We liked this idea because it highlights the strength of deep learning. Captcha images are intentionally designed to "prove you're not a robot" but with deep learning, a robot could solve a captcha.



Features

We used all three of the pixel channels as features. We considered averaging the channels but decided that color differentiation was important in reading the captchas in some case.



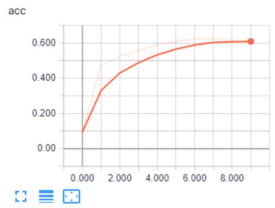
Results

Model	Alphabet Used	String Length	Training Examples	Accuracy Test = 1000	Runtime (mins)	Device
1	0,1	1	10,000	99%	1	CPU
2	0 - 9	1	10,000	99%	1	CPU
2	0,9	2	10,000	81%	5	CPU
2	chars	1	10,000	92%	5	CPU
2	chars	2	100,000	53%	10	CPU
2	chars	3	100,000	41%	15	CPU
2	0,9	3	10,000	65%	10	CPU
2	0-3	6	100,000	85%	15	CPU
2	chars	6	100,000	7%	20	CPU
3	**chars**	**4-6**	**2000**	**96%**	**15**	**GPU**
3	chars	5	2000	99%	12	GPU
3	chars	6	2000	98%	14	GPU
4	chars	4-6	20000	1%	8	GPU
5	chars	4-6	20000	2%	8	GPU
6	chars	4-6	20000	2%	8	AWS
7	chars	4-6	20000	2%	6	AWS
8	chars	4-6	20000	2%	5	AWS

Future

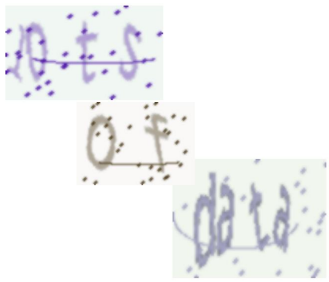
The next step would be to train our model on a bigger server and allow more time for the model to fit our data.

We could also data from other captcha generators. Also we could program an end to end application to use the model in the real world.



Data

Our data comes from a python api used to create captchas. Therefore we can generate more data as we need it. The generator takes a string and image size as input so we can experiment with different lengths, alphabets, and image size.



Model

In our early stages we experimented with fully connected models using relu activation and a sigmoid output layer. in order to examine the effects of data generation in training. We found it was best to keep are large amount of data stored to avoid having to generate during every iteration but then to create more data if desired during the training process.

Model	Description	Language
1	Our First Model - Linear Regression Model for Character Recognition	**TensorFlow**
2	Our Experiment Model - 4 layers of flexible size used to determine effects of activations, data augmentation, accuracy functions, loss functions, and data generation.	**TensorFlow**
3	2 Layer CNN sigmoid added as last layer	**Keras**
4	VGG16 retrained and sigmoid added as last layer	**Keras**
5	VGG19 retrained and sigmoid added as last layer	**Keras**
6	ResNet50 retrained and sigmoid added as last layer	**Keras**
7	Inception V3 retrained and sigmoid added as last layer	**Keras**
8	Xception retrained and sigmoid added as last layer	**Keras**

Discussion

The pixel bug
Generating data
More images vs faster models
Flexibility of TF vs Keras
All in one web extension
Inherent difficulty
Human error solving captchas
Image overlap
Complex models vs Simple solutions
Lack of Computing Power as motivation
Collabratory

References

Jackon Yang. "Captcha Solving Using TensorFlow." Internet: https://github.com/JackonYang/captcha-tensorflow, July 25, 2018 [Mon Dec 3].

CS230 Hands-on session 6: "TensorFlow Blitz with PyTorch Bits" Internet: https://colab.research.google.com/drive/1nilPoE43JgHaBzUHLeUFPx1rtM8uambw#scrollTo=uB6vfKePX53D , July 25, 2018 [Mon Dec 3].