Generative Adversarial Text to Image Synthesis
&
Learning Deep Representations of Fine-Grained Visual Descriptions

this goofy looking, large bird has a bright orange break with a musky gray body and charcoal wing feathers.

AARK KODURU & SPENCER HANN

Generative Adversarial Text to Image Synthesis

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran
Bernt Schiele, Honglak Lee
REEDSCOT[1], AKATA[2], XCYAN[1], LLAJAN[1]
SCHIELE[2],HONGLAK[1]
[1] University of Michigan, Ann Arbor, MI, USA (UMICH.EDU)
[2] Max Planck Institute for Informatics, Saarbrücken, Germany (MPI-INF.MPG.DE)

Abstract

Automatic synthesis of realistic images from text would be interesting and useful, but current AI systems are still far from this goal. However, in recent years generic and powerful recurrent neural network architectures have been developed to learn discriminative text feature representations. Meanwhile, deep convolutional generative adversarial networks (GANs) have begun to generate highly compelling images of specific categories, such as faces, album covers, and room interiors. In this work, we develop a novel deep architecture and GAN formulation to effectively bridge these advances in text and image modeling, translating visual concepts from characters to pixels. We demonstrate the capability of our model to generate plausible images of birds and flowers from detailed text descriptions.

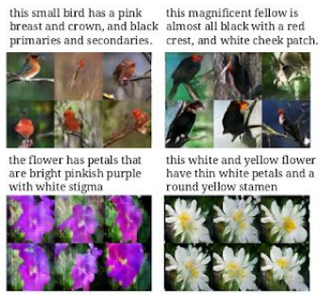


Figure 1. Examples of generated images from text descriptions. Left: captions are from zero-shot (held out) categories, unseen text. Right: captions are from the training set.

1. Introduction

In this work we are interested in translating text in the form of single-sentence human-written descriptions directly into image pixels. For example, "this small bird has a short, pointy orange beak and white belly" or "the petals of this flower are pink and the anther are yellow". The problem of generating images from visual descriptions gained interest in the research community, but it is far from being solved.

Traditionally this type of detailed visual information about an object has been captured in attribute representations - distinguishing characteristics the object category encoded into a vector (Farhadi et al., 2009; Kumar et al., 2009; Parikh & Grauman, 2011; Lampert et al., 2014), in particular to enable zero-shot visual recognition (Fu et al., 2014; Akata et al., 2015), and recently for conditional image generation (Yan et al., 2015).

While the discriminative power and strong generalization properties of attribute representations are attractive, attributes are also cumbersome to obtain as they may require domain-specific knowledge. In comparison, natural language offers a general and flexible interface for describing objects in any space of visual categories. Ideally, we could have the generality of text descriptions with the discriminative power of attributes.

Recently, deep convolutional and recurrent networks for text have yielded highly discriminative and generalizable (in the zero-shot learning sense) text representations learned automatically from words and characters (Reed et al., 2016). These approaches exceed the previous state-of-the-art using attributes for zero-shot visual recognition on the Caltech-UCSD birds database (Wah et al., 2011), and also are capable of zero-shot caption-based retrieval. Motivated by these works, we aim to learn a mapping directly from words and characters to image pixels.

To solve this challenging problem requires solving two sub-problems: first, learn a text feature representation that captures the important visual details; and second, use these fea-

Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

Generative Adversarial Text to Image Synthesis
Scott Reed, et. al
https://arxiv.org/pdf/1605.05396.pdf

Learning Deep Representations of Fine-Grained Visual Descriptions

Scott Reed[1], Zeynep Akata[2], Honglak Lee[1] and Bernt Schiele[2]
[1]University of Michigan [2]Max-Planck Institute for Informatics

Abstract

State-of-the-art methods for zero-shot visual recognition formulate learning as a joint embedding problem of images and side information. In these formulations the current best complement to visual features are attributes: manually-encoded vectors describing shared characteristics among categories. Despite good performance, attributes have limitations: (1) finer-grained recognition requires commensurately more attributes, and (2) attributes do not provide a natural language interface. We propose to overcome these limitations by training neural language models from scratch; i.e. without pre-training and only consuming words and characters. Our proposed models train end-to-end to align with the fine-grained and category-specific content of images. Natural language provides a flexible and compact way of encoding only the salient visual aspects for distinguishing categories. By training on raw text, our model can do inference on raw text as well, providing humans a familiar mode both for annotation and retrieval. Our model achieves strong performance on zero-shot text-based image retrieval and significantly outperforms the attribute-based state-of-the-art for zero-shot classification on the Caltech-UCSD Birds 200-2011 dataset.

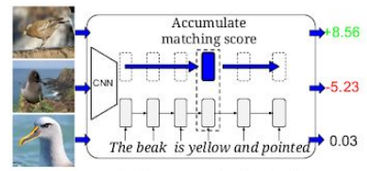


Figure 1: Our model learns a scoring function between images and text descriptions. A word-based LSTM is shown here, but we also evaluate several alternative models.

1. Introduction

A key challenge in image understanding is to correctly relate natural language concepts to the visual content of images. In recent years there has been significant progress in learning visual-semantic embeddings, *e.g.* for zero-shot learning [36, 38, 24, 33, 12, 41, 2] and automatically generating image captions for general web images [23, 35, 45, 20, 8]. These methods have harnessed large image and text datasets [39, 50, 25], as well as advances in deep neural networks for image and language modeling, already enabling powerful new applications such as auto-captioning images for blind users on the web [27].

Despite these advances, the problem of relating images and text is still far from solved. In particular for the fine-grained regime [46, 10, 7, 51], where images of different classes have only subtle distinctions, sophisticated language models have not been employed, perhaps due to the scarcity of large and high-quality training data. For instance on the Caltech-UCSD birds database (CUB) [46], previous zero-shot learning approaches [13, 2, 3] have used human-encoded attributes [24], or simplified language models such as bag-of-words [16], WordNet-hierarchy-derived features [29], and neural word embeddings such as Word2Vec [28] and GloVE [37].

Previous text corpora used for fine-grained label embedding were either very large but not visually focused, *e.g.* the entire `wikipedia`, or somewhat visually relevant but very short, *e.g.* the subset of `wikipedia` articles that are related to birds. Furthermore, these wikis do not provide enough aligned images and text to train a high-capacity sentence encoder. Given the data limitations, previous text embedding methods work surprisingly well for zero-shot visual recognition, but there remains a large gap between the text embedding methods and human-annotated attributes (28.4% vs 50.1% average top-1 per-class accuracy on CUB [2]).

In order to close the performance gap between text embeddings and human-annotated attributes for fine-grained visual recognition, we hypothesize that higher-capacity text models are required. However, more sophisticated text models would in turn require more training data, in particular aligned images and multiple visual descriptions per image for each fine-grained category. These descriptions

Learning Deep Representations of Fine-Grained Visual Descriptions
Scott Reed, et. al
https://arxiv.org/pdf/1605.05395.pdf

INITIAL FEEDBACK

- I’m a bit concerned that this project is too ambitious for a 3-week class project. This is a fascinating but possibly overambitious proposal. - Melanie Mitchell

Dear Spencer, Aark, and Jeffrey,

This is a fascinating but possibly overambitious proposal. A few questions:

What platform / library are you going to use to build the GAN?

How are you going to ensure enough compute power to do the training?

I’m a bit concerned that this project is too ambitious for a 3-week class project. Any ideas for simplifying it? E.g., using CNNs to do classification on these datasets rather than dealing with GANs?

Melanie

PROJECT ASPECTS

- Machine Learning
    - Naive GAN
    - GAN INT
    - GAN CLS
- Natural language processing
    - Char CNN-RNN
    - Word2Vec with stop words
    - Word2Vec without stop words

Dataset: Caltech-UCSD Birds-200-2011

- 11,788 images of birds
- 30 attribute types
- 312 total attributes
- etc.

Caltech-UCSD Birds-200-2011

Warning: Images in this dataset overlap with images in ImageNet. Exercise caution when using networks pretrained with ImageNet (or any network pretrained with images from Flickr) as the test set of CUB may overlap with the training set of the original network.

Browse

Click here to browse the dataset.

Details

Caltech-UCSD Birds-200-2011 (CUB-200-2011) is an extended version of the CUB-200 dataset, with roughly double the number of images per class and new part location annotations. For detailed information about the dataset, please see the technical report linked below.

- **Number of categories:** 200
- **Number of images:** 11,788
- **Annotations per image:** 15 Part Locations, 312 Binary Attributes, 1 Bounding Box

DATA SET

Attributes: Discrete Nominal values in Discrete Nominal Categories

1 has_bill_shape::curved_(up_or_down)
2 has_bill_shape::dagger
3 has_bill_shape::hooked
4 has_bill_shape::needle
5 has_bill_shape::hooked_seabird
6 has_bill_shape::spatulate
7 has_bill_shape::all-purpose
8 has_bill_shape::cone
9 has_bill_shape::specialized
10 has_wing_color::blue
11 has_wing_color::brown
12 has_wing_color::iridescent
...
92 has_upper_tail_color::white
93 has_upper_tail_color::red
94 has_upper_tail_color::buff
95 has_head_pattern::spotted
96 has_head_pattern::malar
97 has_head_pattern::crested
98 has_head_pattern::masked
99 has_head_pattern::unique_pattern
100 has_head_pattern::eyebrow

...
211 has_belly_color::red
212 has_belly_color::buff
213 has_wing_shape::rounded-wings
214 has_wing_shape::pointed-wings
215 has_wing_shape::broad-wings
216 has_wing_shape::tapered-wings
217 has_wing_shape::long-wings
218 has_size::large_(16_-_32_in)
219 has_size::small_(5_-_9_in)
220 has_size::very_large_(32_-_72_in)
221 has_size::medium_(9_-_16_in)
222 has_size::very_small_(3_-_5_in)
223 has_shape::upright-perching_water-like
224 has_shape::chicken-like-marsh
225 has_shape::long-legged-like
226 has_shape::duck-like
...
309 has_wing_pattern::solid
310 has_wing_pattern::spotted
311 has_wing_pattern::striped
312 has_wing_pattern::multi-colored

DATA SET

Attributes: Discrete Nominal values in Discrete Nominal Categories

- Easier for computers to understand
- Manually generated by humans
- Better feature recognition requires more and more attribute types
- No natural language interface

DATA SET

- Lots of datasets w/images (of Birds)
  - Images need to have *paired* description(s), not just discrete attributes/features
- Lots of large corpora
  - May not have “visual” language need to learn visual concepts from text
- CUB-200 had image/Bird attributes, but not text descriptions
  - Amazon Mechanical Turk
    - Visual descriptions of at least 10 words
    - Avoid figurative language
    - Never explicitly state species even if known

Modified Caltech-UCSD Birds(CUB-200-2011)

- Number of categories: 200
- 10 descriptions per Image
- Number of images: 11,788
- Number of image-description pairs: 117,880
- Word count with stopwords: 1,996,042
- Vocabulary size with stopwords: 6,888
- Word count without stopwords: 1,220,898
- Vocabulary size without stopwords: 6,774

Sample from the dataset

Descriptions:

- this small bird has a black eye ring and throat, a red crown, and light grey an tan feathers covering most of its body.
- this bird is white, brown and black in color, with a black beak.

Bohemian Waxwing

Sample from the dataset

Descriptions:

- this goofy looking, large bird has a bright orange break with a musky gray body and charcoal wing feathers.
- this bird is grey in color with a vibrant orange beak, and white eye rings.
- this bird has a hairy crown, a gray belly, breast, throat, tarsus & feet, and a bright orange area surrounding its bill.

Crested Auklet

Samples from the dataset

Generative Adversarial Networks Primer

Generative Adversarial Networks Primer

Discriminator Network

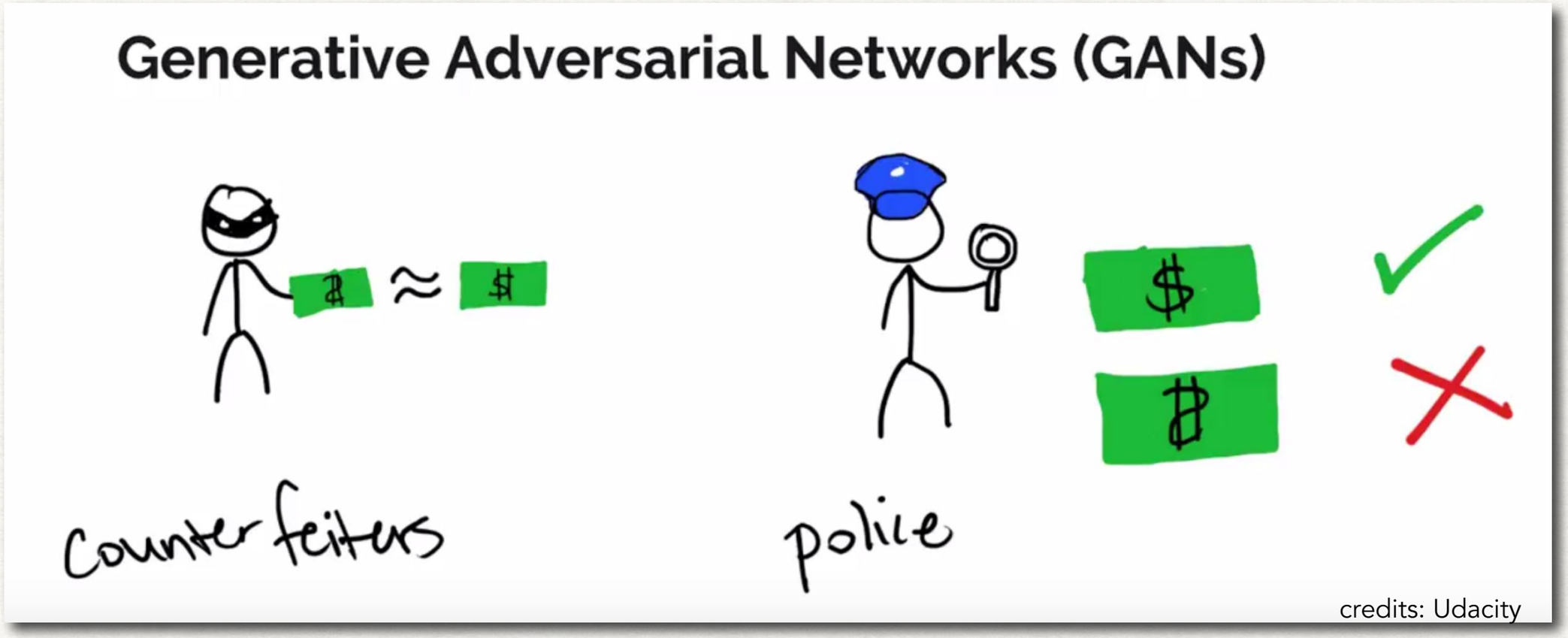
- neural classifier
- classifies examples as legitimate or fake

Generator Network

- “backwards” neural classifier
- forward propagates in reverse to produce examples from class labels

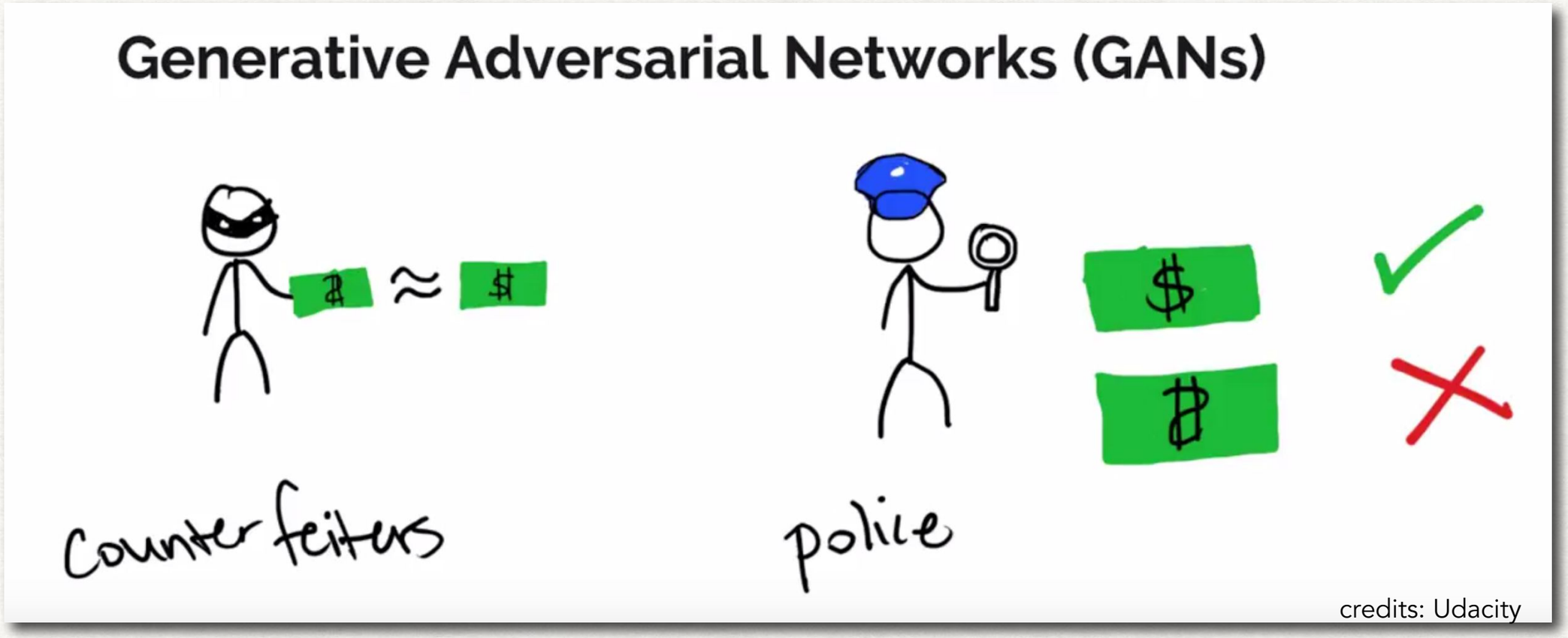
* Epoch one

HOW GANS WORK



credits: Udacity

HOW GANS WORK

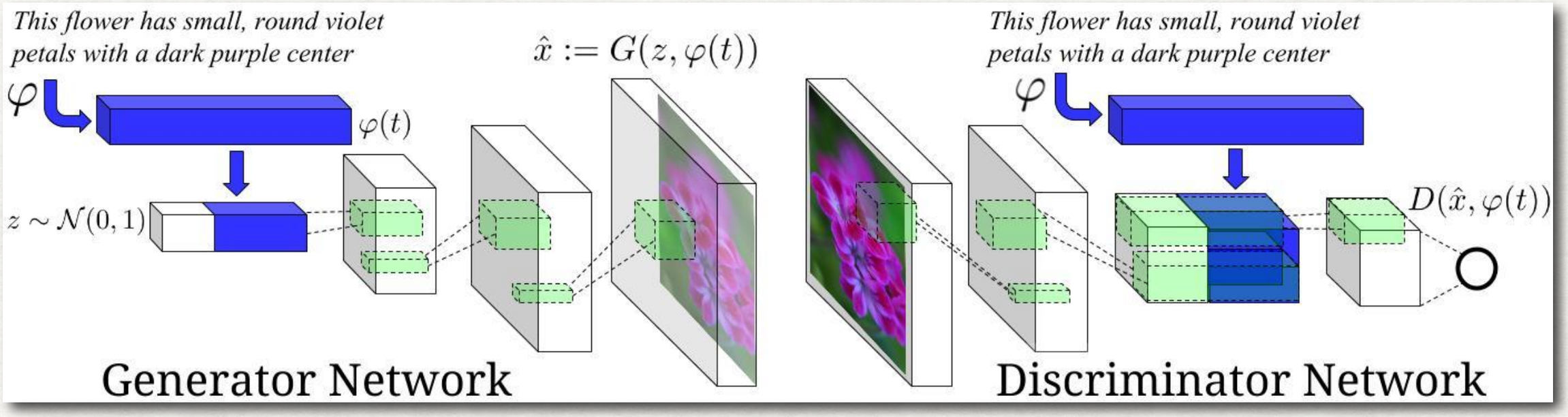


$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data x}} + \mathbb{E}_{z \sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data G(z)}}) \right]$$

Source: [Generative Adversarial Text-to-Image Synthesis](#) Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee

ARCHITECTURE

NAIVE GENERATIVE ADVERSARIAL NETWORK(GAN)



$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data x}} + \mathbb{E}_{z \sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data G(z)}}) \right]$$

Source: [Generative Adversarial Text-to-Image Synthesis](#) Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee

Text Encodings

Word Embeddings

Text Encodings

Word Embeddings

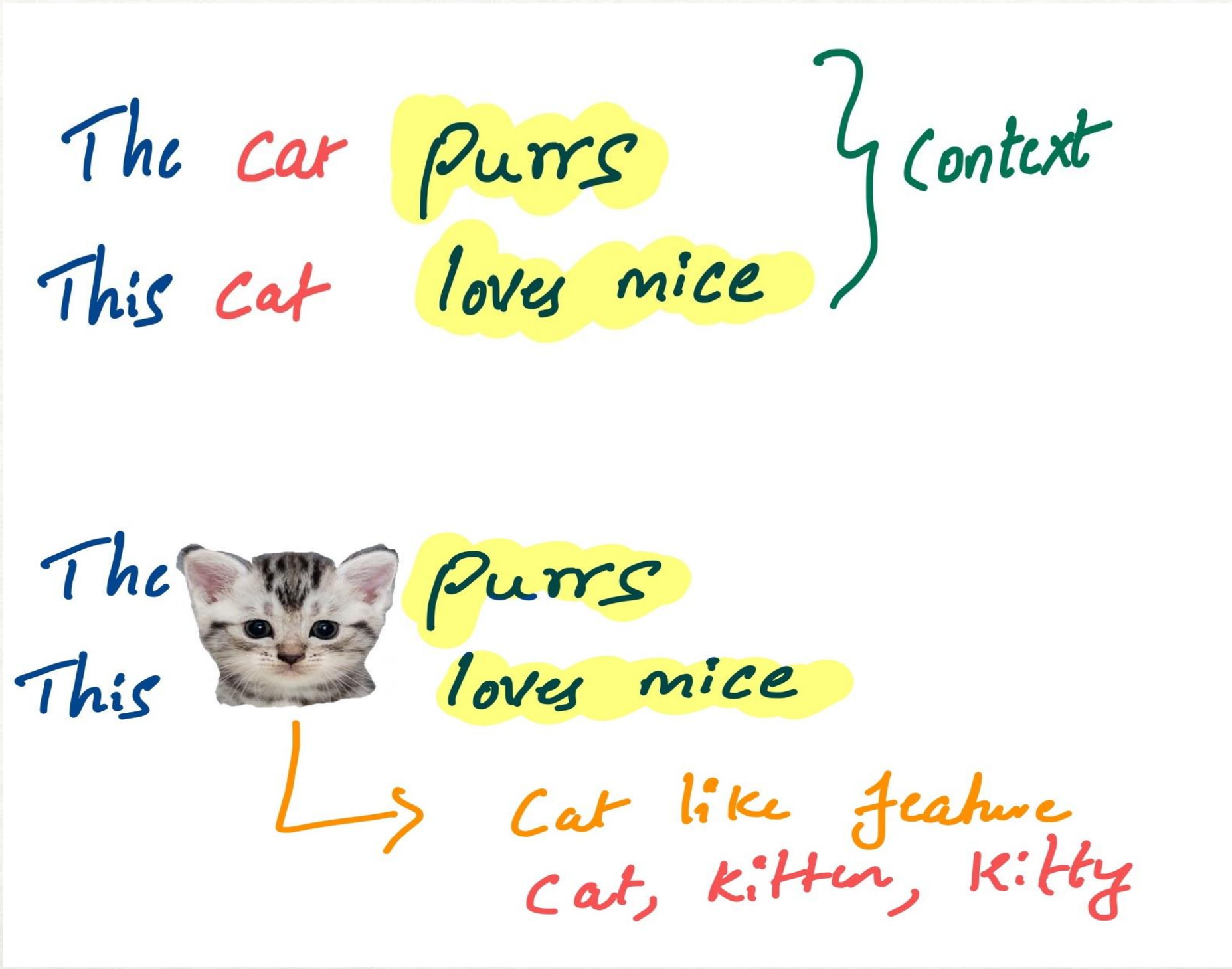
- Word embeddings are words mapped to vectors of real numbers
- Word2vec is not a single algorithm but a combination of two techniques – CBOW(Continuous bag of words) and Skip-gram model.
- It places words in vector space which has some semantic meaning
- Most of the word embedding algorithms learn some sort of weights which act as word vector representations

Text Encodings

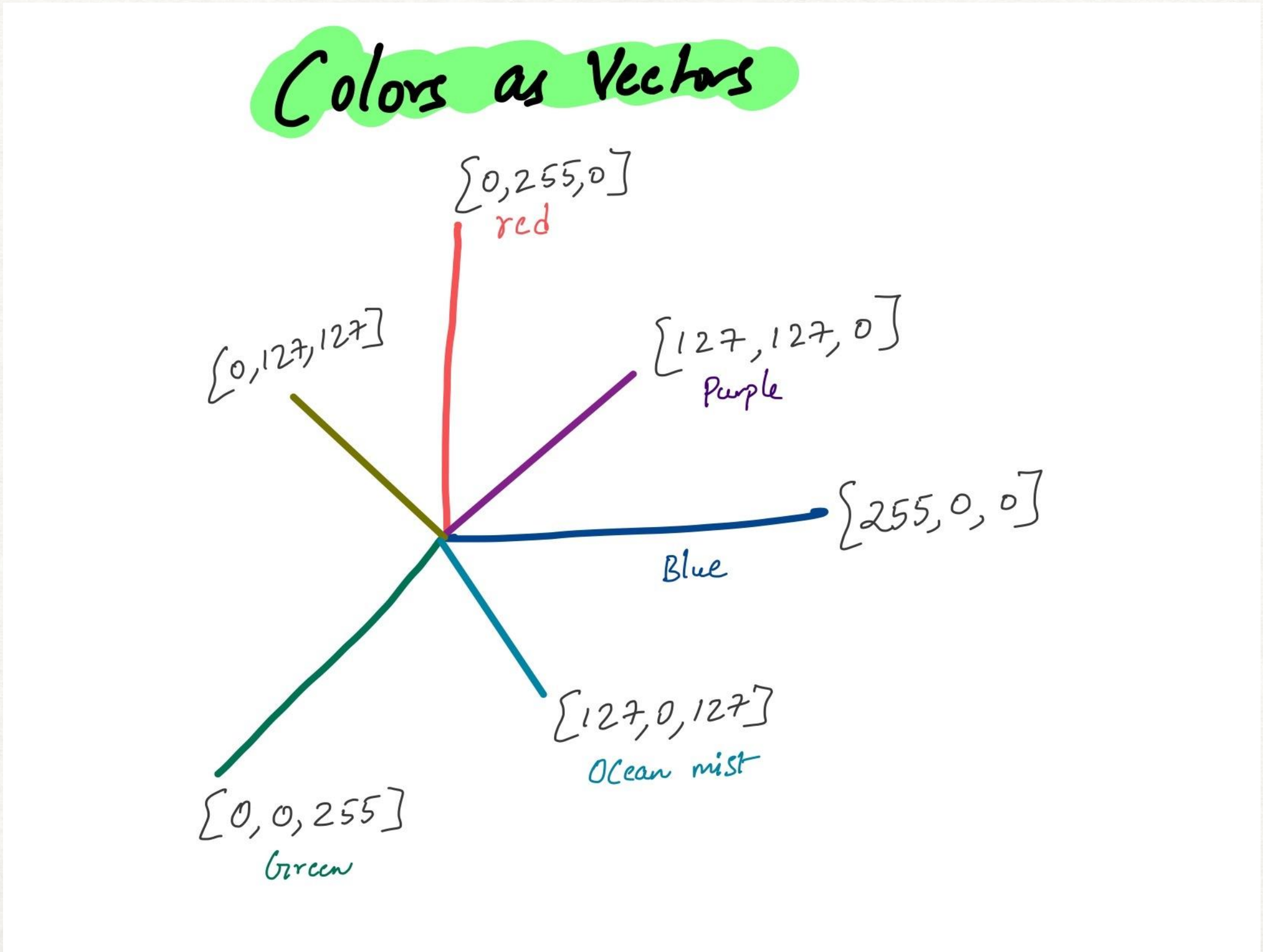
Word Embeddings

Text Encodings

Word Embeddings



Colors (RGB) analogy for word embeddings

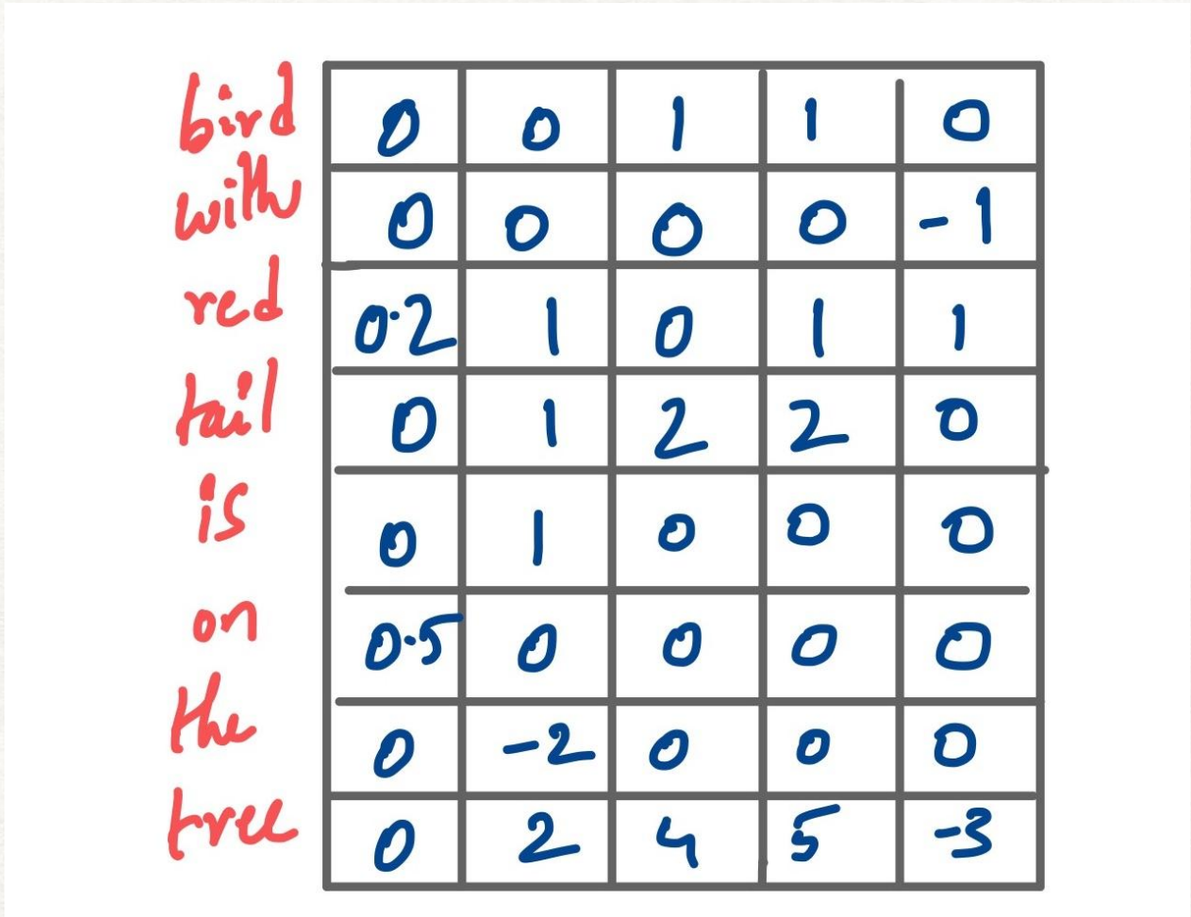


RGB analogy for word embeddings

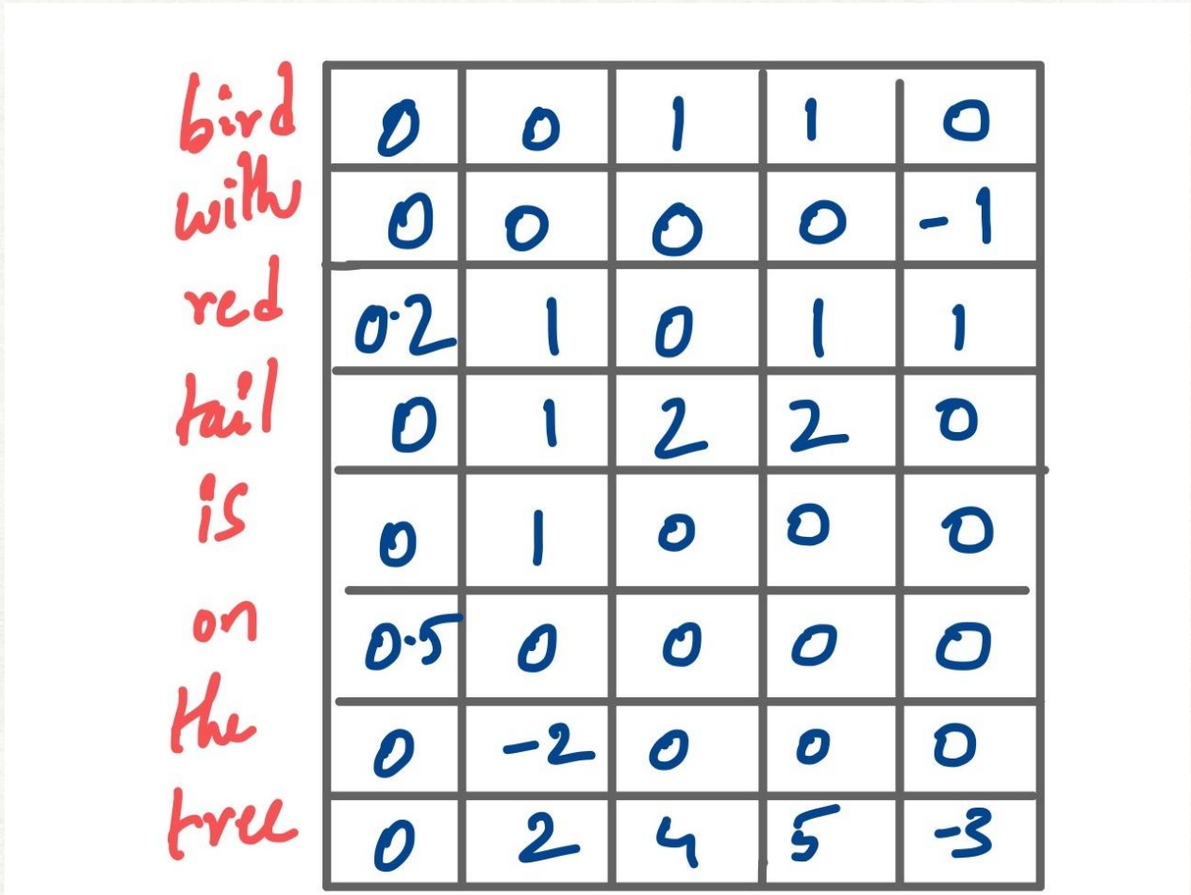
PURPLE - **RED** = **BLUE**

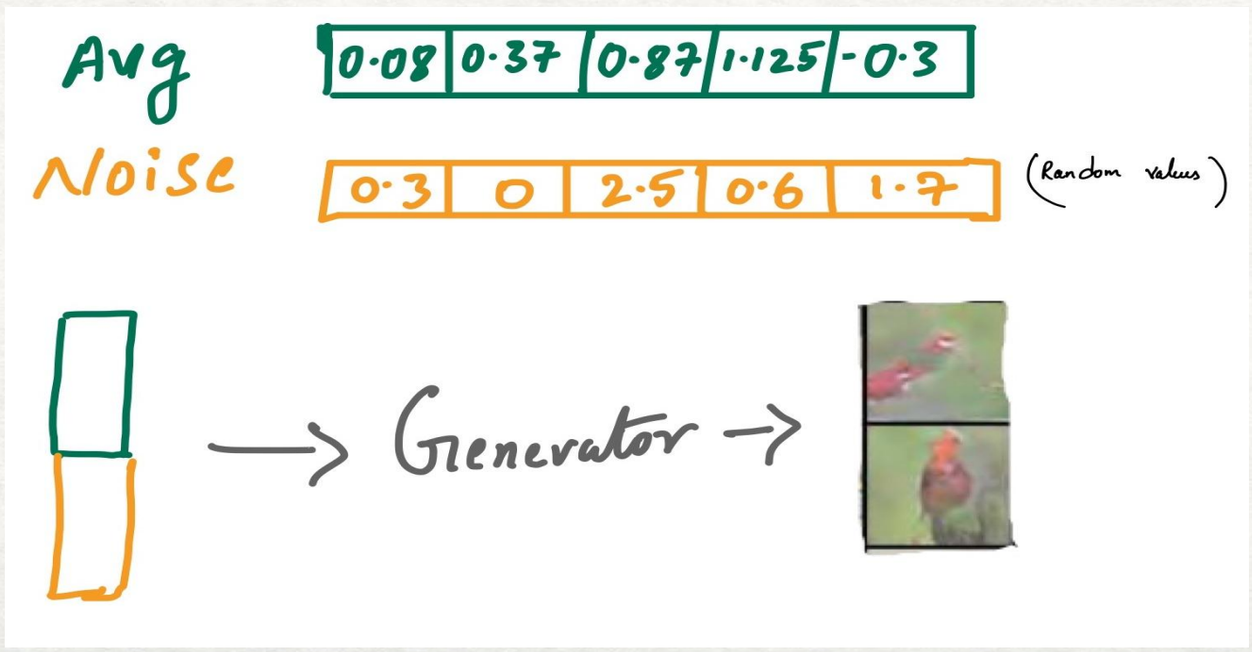
[255,0,255] - [255,0,0] = [0,0,255]

ARCHITECTURE

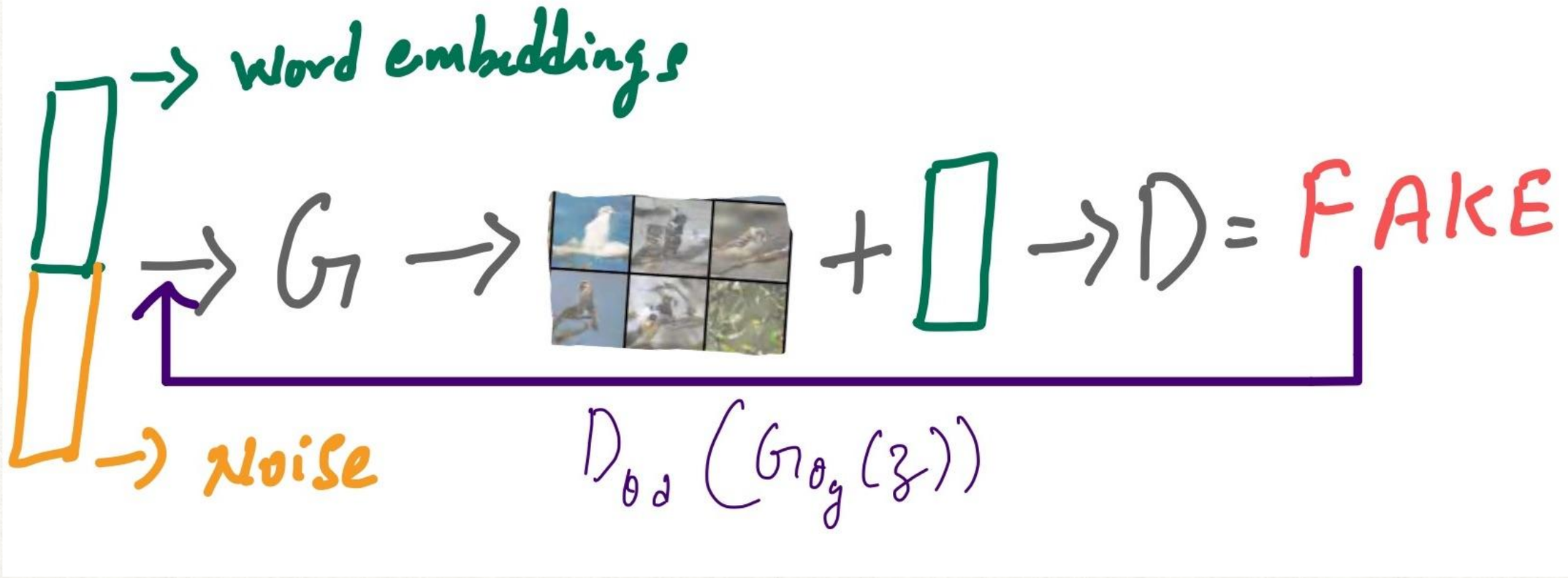


ARCHITECTURE

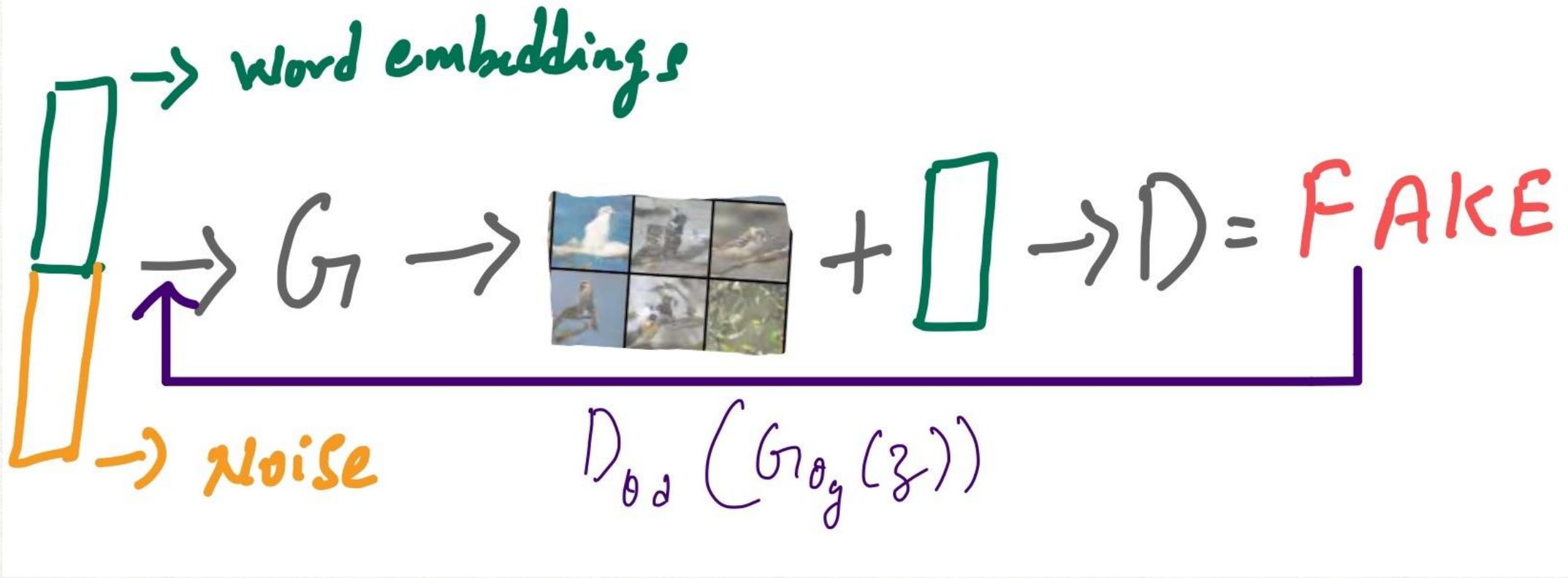




ARCHITECTURE



ARCHITECTURE



Text Encoding
Deep Structured Joint Embeddings

Text Encoding
Deep Structured Joint Embeddings

text descriptions: $t \in T$

visual information: $v \in V$

class labels: $y \in Y$

$$f_t : T \to Y$$

$$f_v : V \to Y$$

$$\Delta : Y \times Y \to \{0, 1\}$$

$$\frac{1}{N} \sum_{n=1}^{N} \Delta(y_n, f_v(v_n)) + \Delta(y_n, f_t(t_n))$$

Deep Structured Joint Embeddings

Compatibility Function

text encoder: φ

image encoder: θ

$$F : V \times T \rightarrow \mathbb{R}$$

$$F(v,t) = \theta(v)^T \varphi(t)$$

Learning Classifier Functions

$$f_t(t) = \arg\max_y \mathbb{E}_{v \sim V(y)}[F(v,t)], \text{ for } y \text{ in } Y$$

$$f_v(v) = \arg\max_y \mathbb{E}_{t \sim T(y)}[F(v,t)], \text{ for } y \text{ in } Y$$

Deep Structured Joint Embeddings

Surrogate Loss Function

$$\frac{1}{N}\sum_{n=1}^{N} l_v(v_n, t_n, y_n) + l_t(v_n, t_n, y_n)$$

$$l_v(v_n, t_n, y_n) =$$
$$max_y(0, \Delta(y_n, y) + \mathbb{E}_{t\sim T(y)}[F(v_n, t) - F(v_n, t_n)])$$

$$l_t(v_n, t_n, y_n) =$$
$$max_y(0, \Delta(y_n, y) + \mathbb{E}_{v\sim V(y)}[F(v, t_n) - F(v_n, t_n)])$$

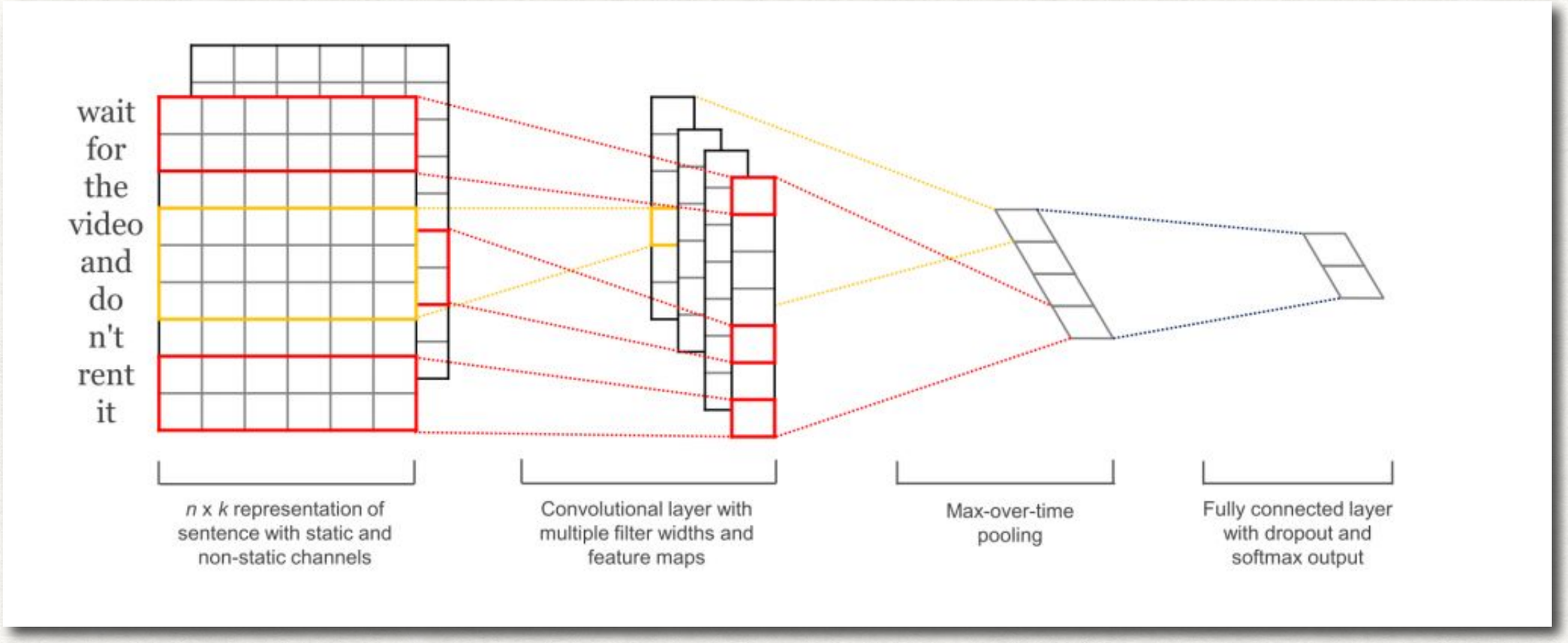
Generating Text ~~(and image)~~ encodings

How?

$$\varphi(t) = \text{neural black magic vector}$$

Generating Text Encodings

WORD CNN

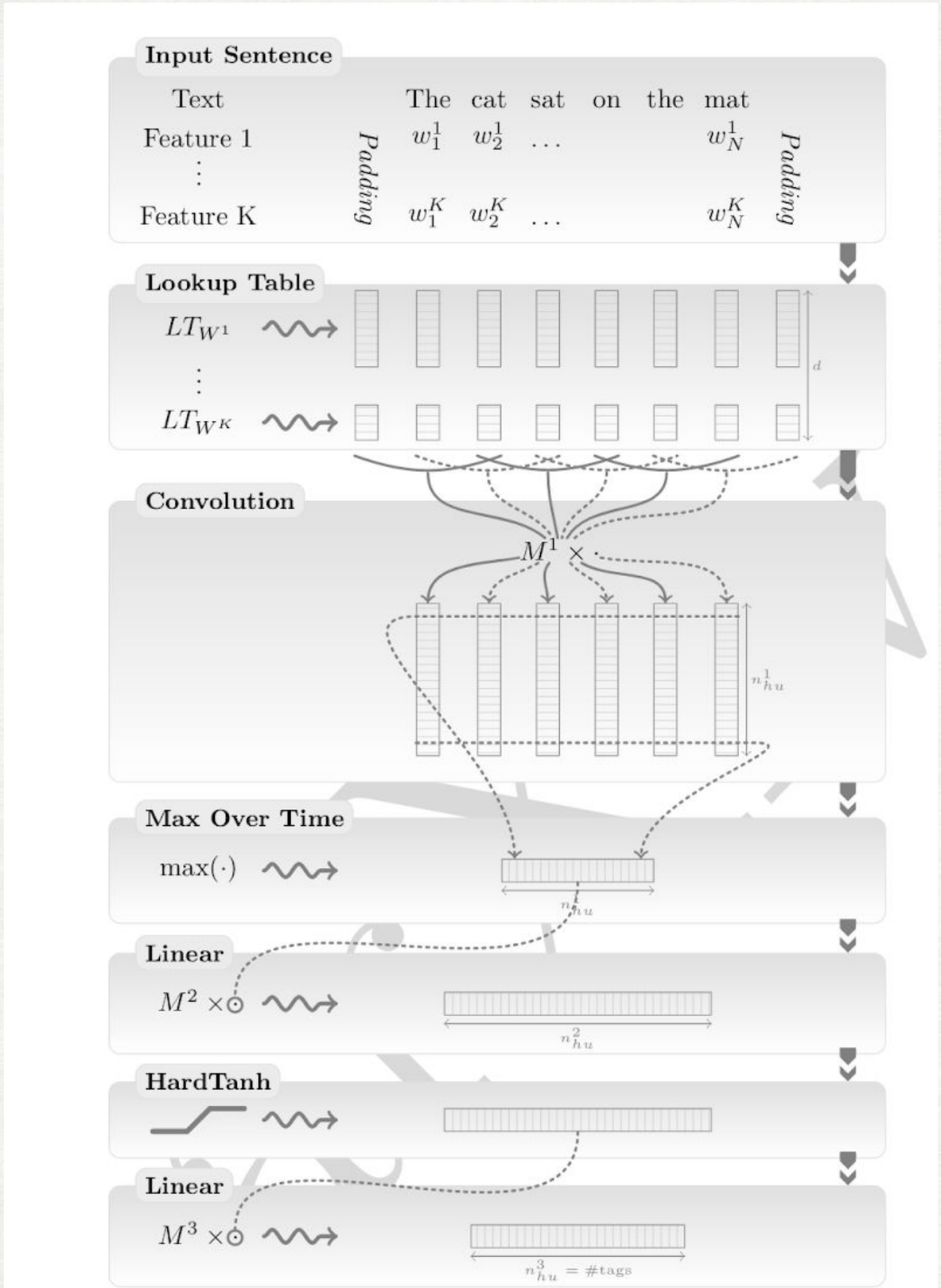


- It's like the typical CNN for image processing
  - pixel width = 1
  - channel = vocabulary/alphabet size

- The convolutional layer convolves over a span for words instead of pixels

Convolutional Neural Networks for Sentence Classification, Yoon Kim, et. al
https://arxiv.org/pdf/1408.5882.pdf

WORD CNN



Natural Language Processing (almost) from Scratch
Ronan Collobert et. al
https://arxiv.org/pdf/1103.0398.pdf

* *relu* instead of *tanh*

Generating Text Encodings

WORD (or CHAR) CNN

$$\varphi(t) = \mathrm{CNN}(t)$$

Generating Text Encodings

Word CNN-RNN

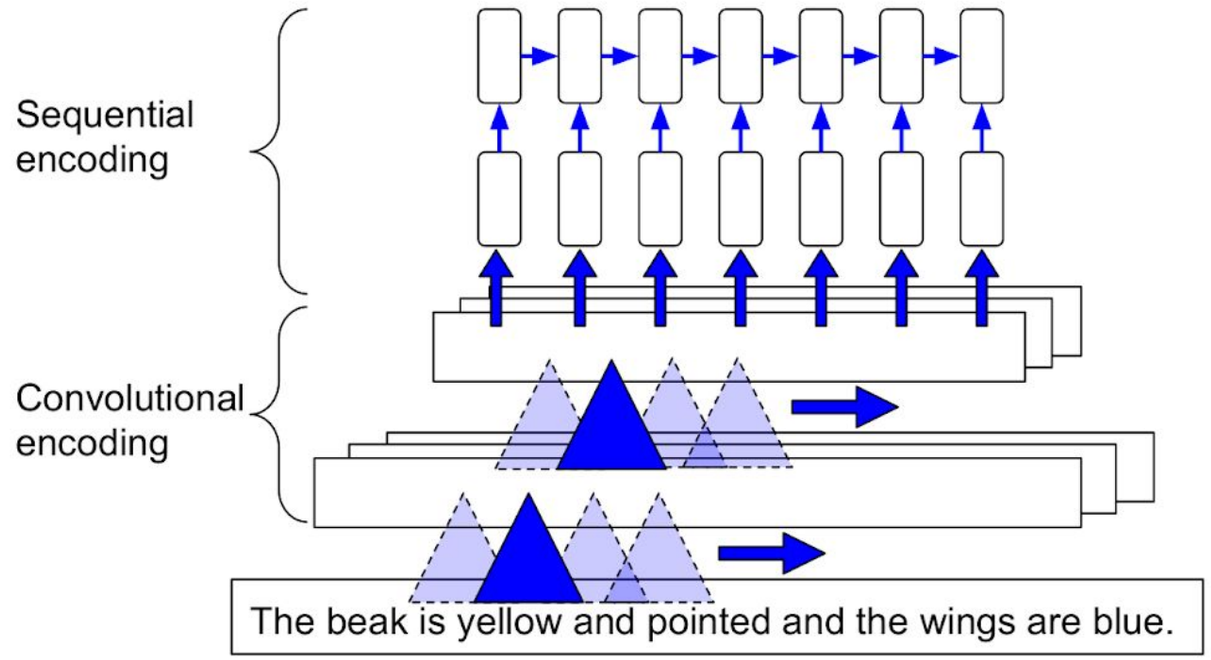


Figure 2: Our proposed convolutional-recurrent net.

Final CNN hidden layer is used as input vector sequence fed into an RNN encoder

Encodes sequential/temporal information that a CNN alone ignores

Generating Text Encodings

CNN-RNN

$$h = \mathrm{RNN}(\mathrm{CNN}(t))$$

$$\varphi(t) = \frac{1}{L}\sum_{i=1}^{L} h_i$$

Generating Text Encodings

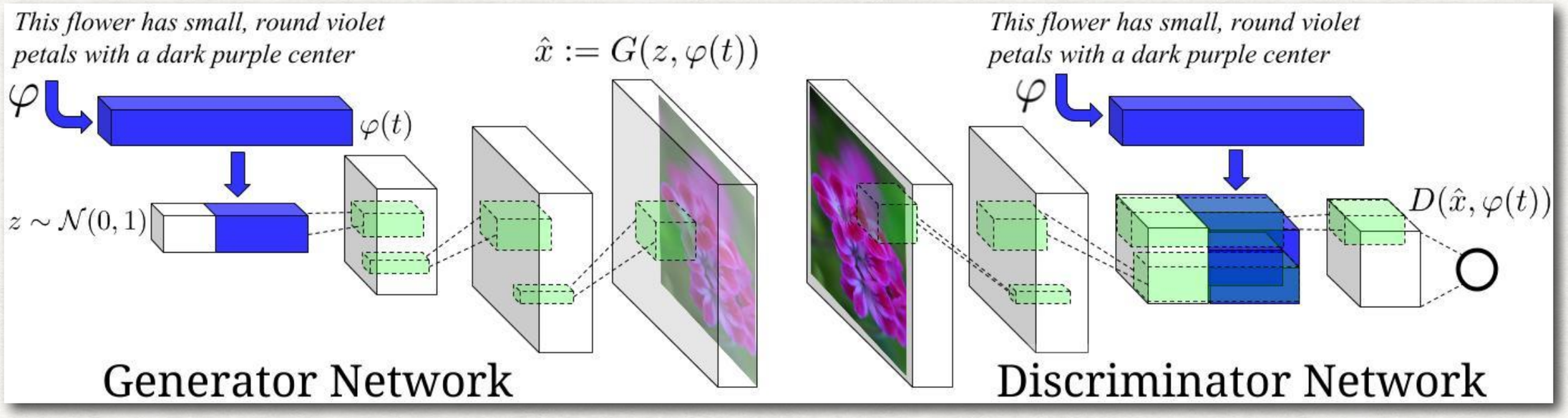
LSTM

$$h = \mathrm{LSTM}(t)$$

$$\varphi(t) = \frac{1}{L} \sum_{i=1}^{L} h_i$$

ARCHITECTURE

NAIVE GENERATIVE ADVERSARIAL NETWORK(GAN)



$$\min_{\theta_g} \max_{\theta_d} \left[\mathbb{E}_{x \sim p_{data}} \log \underbrace{D_{\theta_d}(x)}_{\text{Discriminator output for real data x}} + \mathbb{E}_{z \sim p(z)} \log(1 - \underbrace{D_{\theta_d}(G_{\theta_g}(z))}_{\text{Discriminator output for generated fake data G(z)}}) \right]$$

Source: [Generative Adversarial Text-to-Image Synthesis](#) Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee

Experiments & Results

- GANs are difficult to evaluate
  - Qualitative
  - Subjective

- No consensus as to best way to evaluate

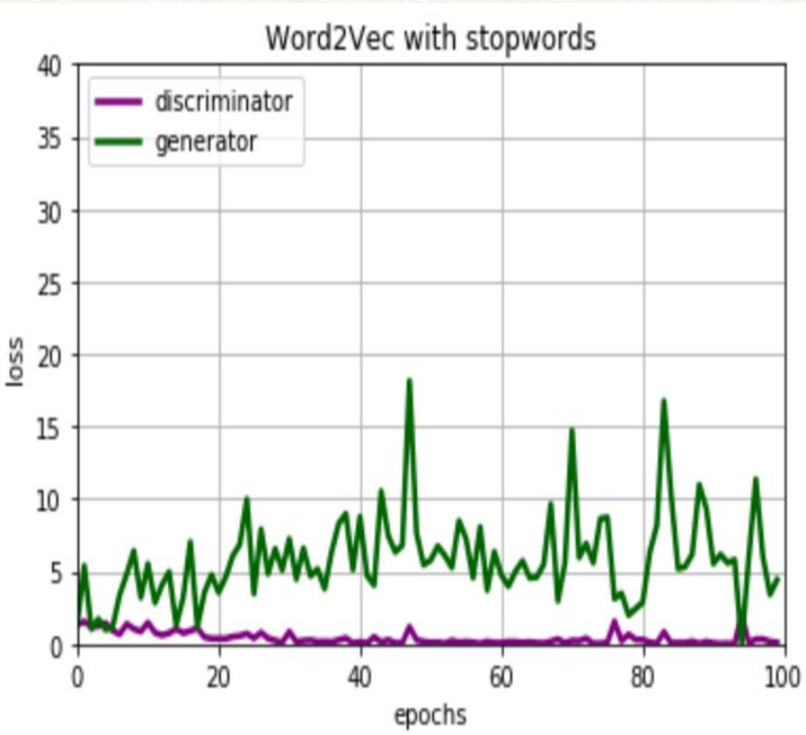
Experiments & Results

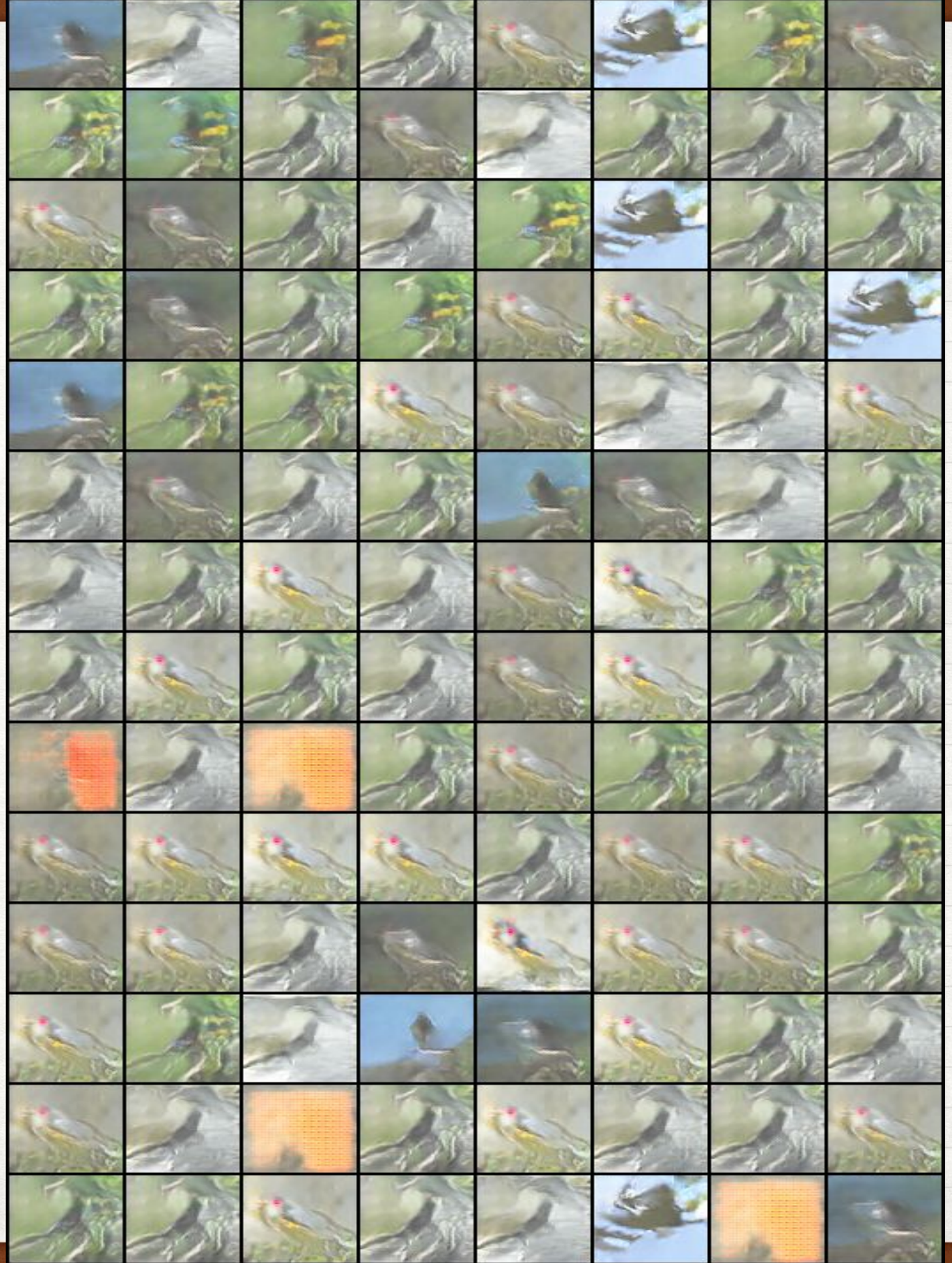
Hyper parameters:

- Total no. of epochs: 100
  - Original research used 1000
- Learning rate: 0.0004
  - Original research used 0.0002
- beta: 0.3
  - Original research used 0.5
- batch size: 128
  - Original research used 64
- Pre-trained word embeddings of size 1024
  - Original research used 1024 size vectors

Word2Vec

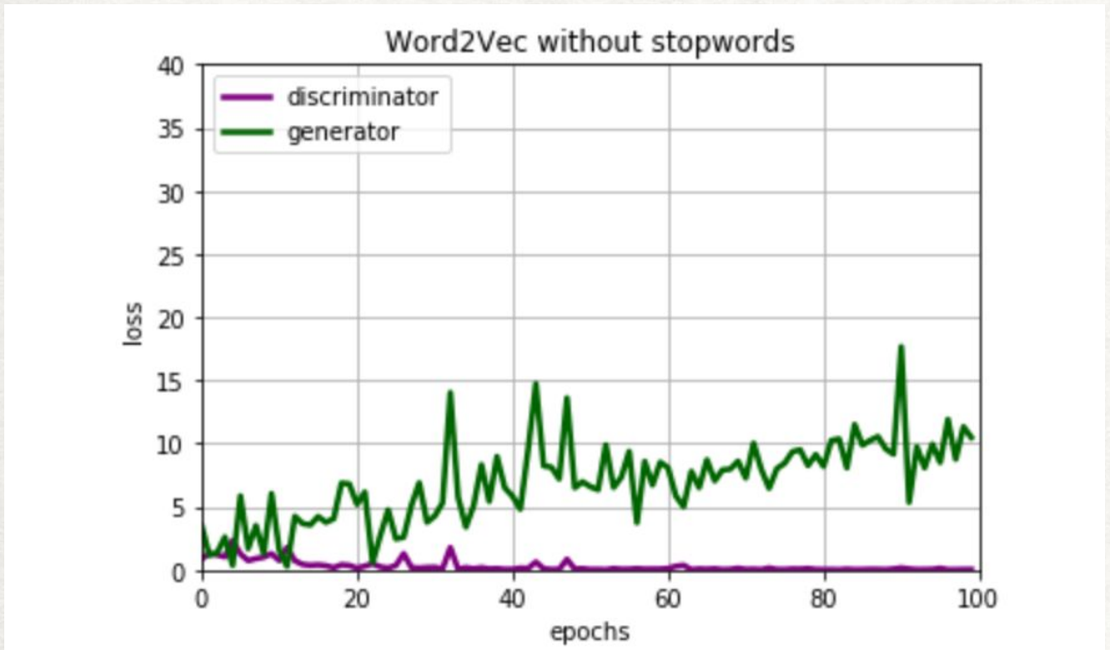
with stopwords



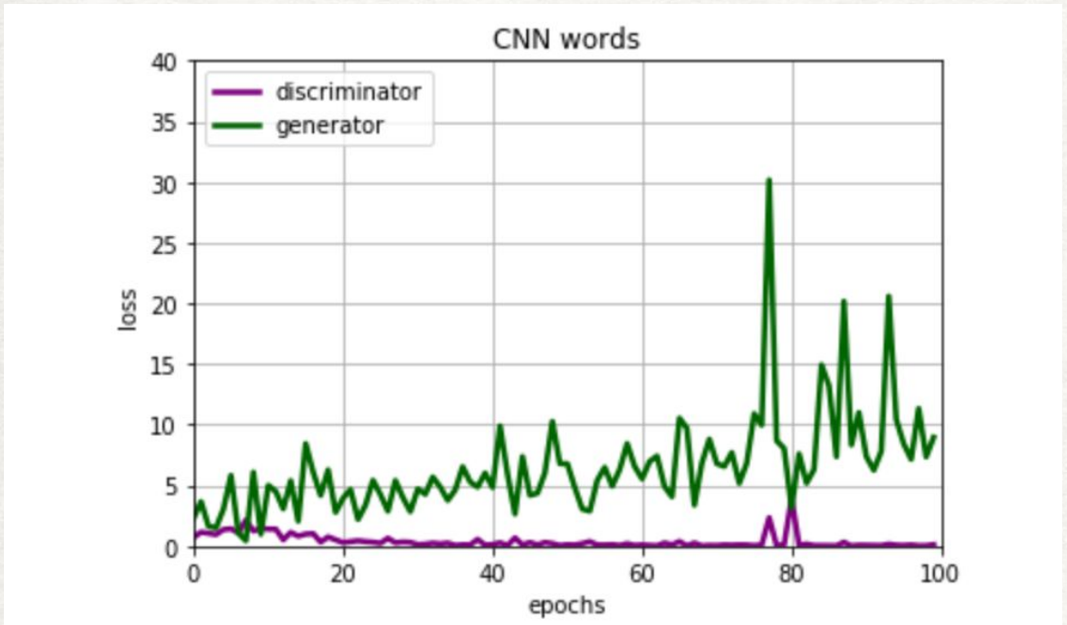


Word2Vec

without stop words



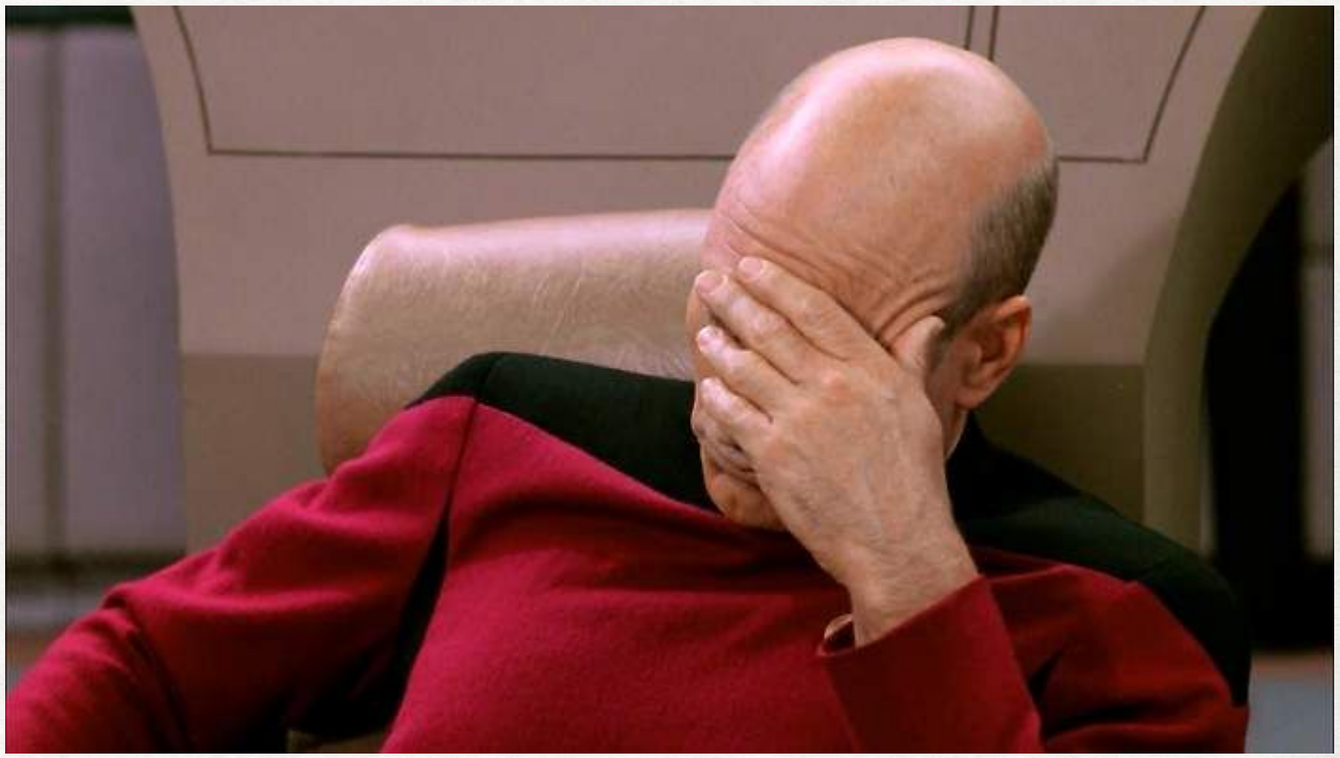
Char CNN-RNN


CNN words
discriminator
generator
loss
epochs
0
5
10
15
20
25
30
35
40
0
20
40
60
80
100

GLOVE

- Nope.
- Trained for 6 hours before crashing
  - not sure what happened
- Failed Experiment

You can’t cram the whole $#@*ing project and try to finish it in a single ^*!%ing week



FUTURE EXPERIMENTS

- Train word embeddings on domain specific corpus like ornithology texts or wiki pages of birds
- Create text encodings by averaging across all descriptions
- Use weights to import the significance of features like color, shape etc. while averaging the word embeddings
    - K-means clustering around predefined attributes
- Use more sophisticated GANs like GAN-INT with the word embeddings
- Experiment with Word2Vec hyperparameters, i.e. window size
    - Experiment with skip-gram word embeddings

FUTURE EXPERIMENTS

- Train Word Embeddings on large non-domain specific corpus
  - test for zero-shot learning
- Smaller batch sizes
- More descriptions per images
- Longer training times (600 epochs as opposed to 100)
- Use Inception network as a metric for evaluating Generator's images
- Experiment with different beta values for interpolation

Demo

"They don't appear to want to take over. They just want to dance."

credits: The New Yorker