deeplearning.ai

Conditional Generation: Intuition

Outline

- Unconditional generation
- Conditional vs. unconditional generation

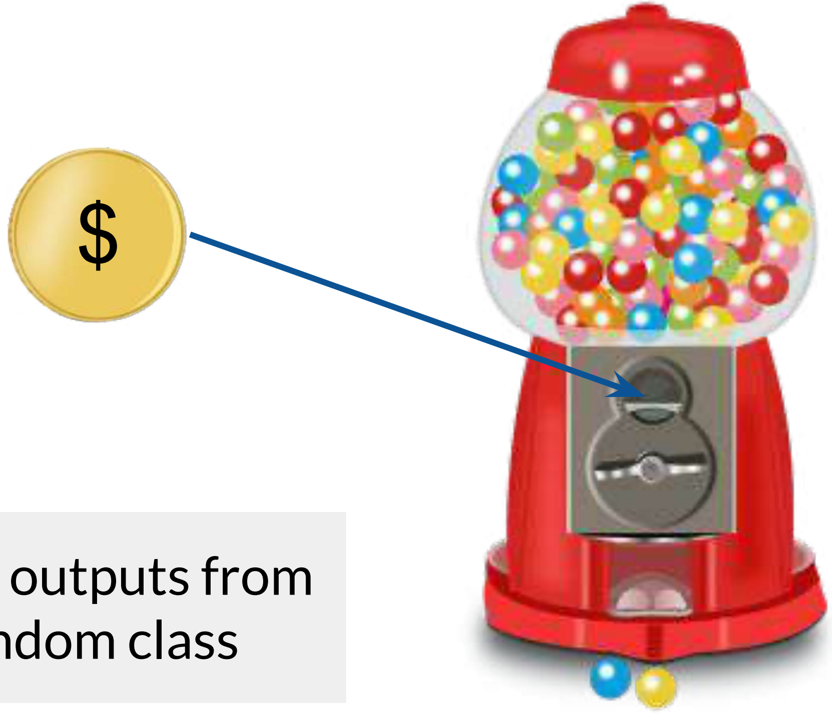
Unconditional Generation

You get outputs from a random class

Unconditional Generation

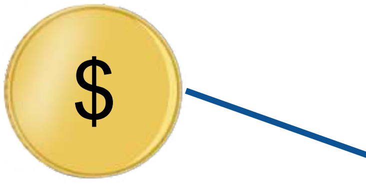
You get outputs from a random class

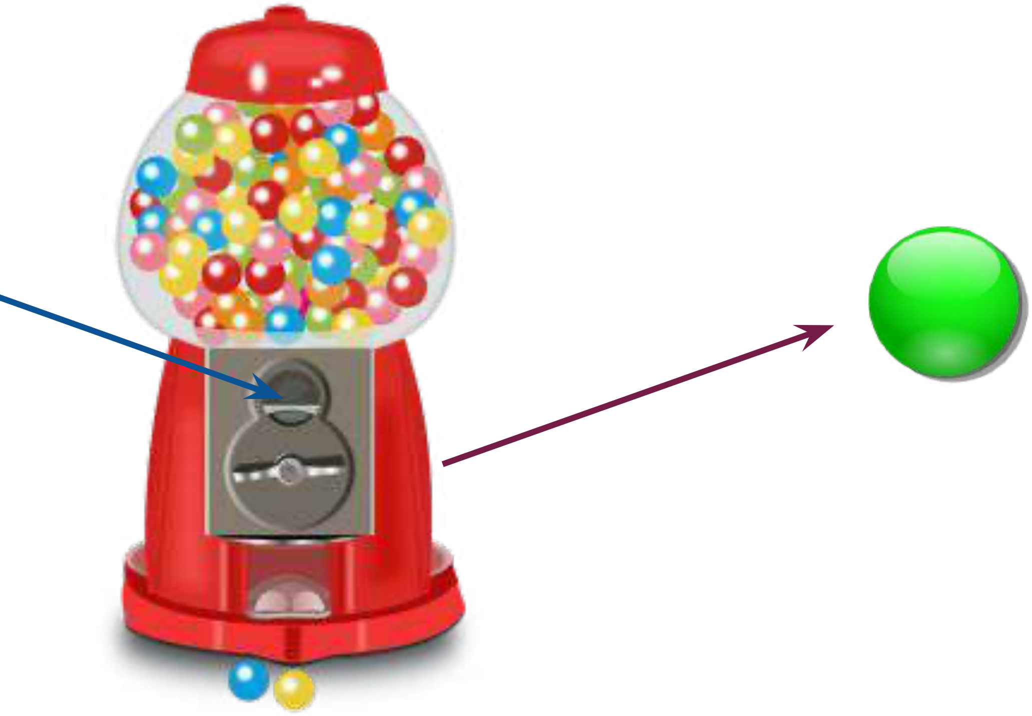
Unconditional Generation



You get outputs from a random class

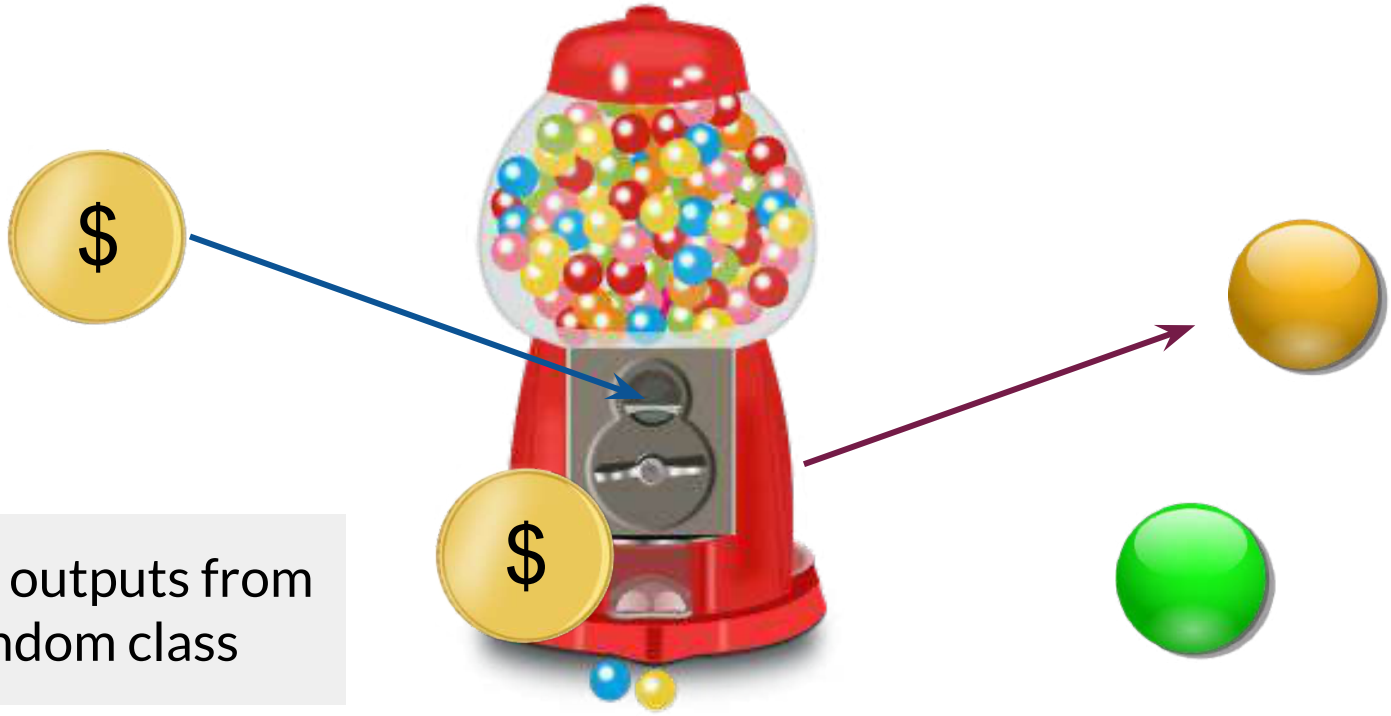
Unconditional Generation





You get outputs from a random class

Unconditional Generation



You get outputs from a random class

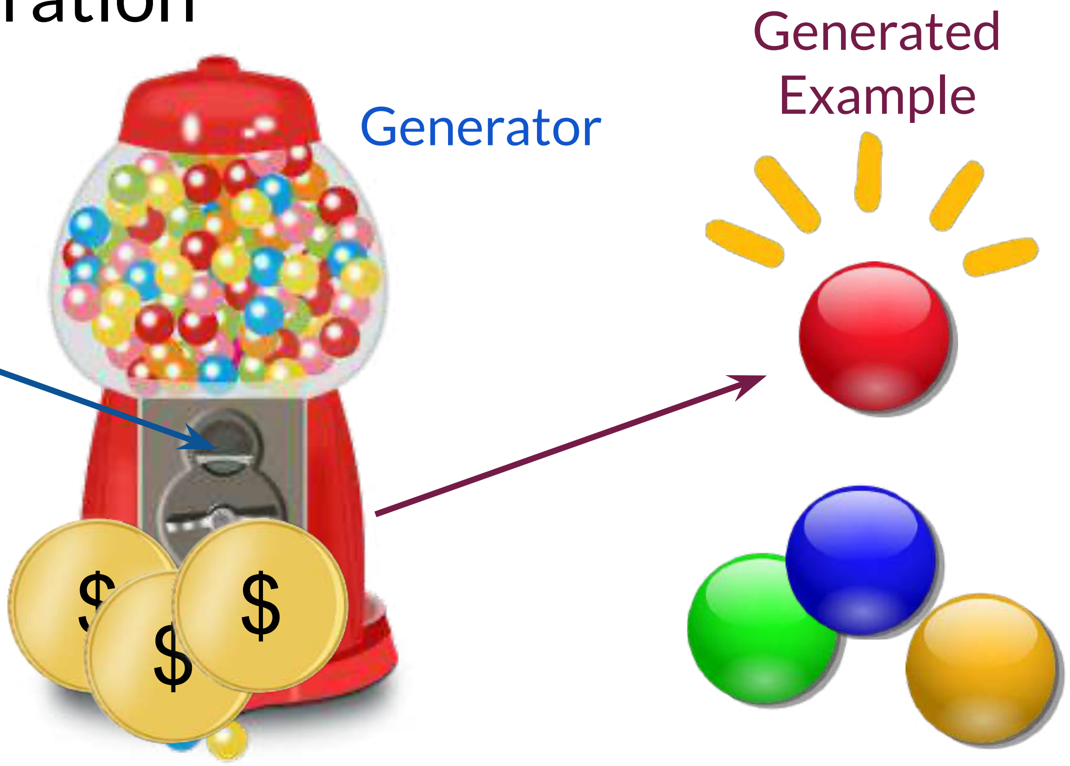
Unconditional Generation

You get outputs from a random class

Unconditional Generation

You get outputs from a random class

Unconditional Generation



You get outputs from a random class

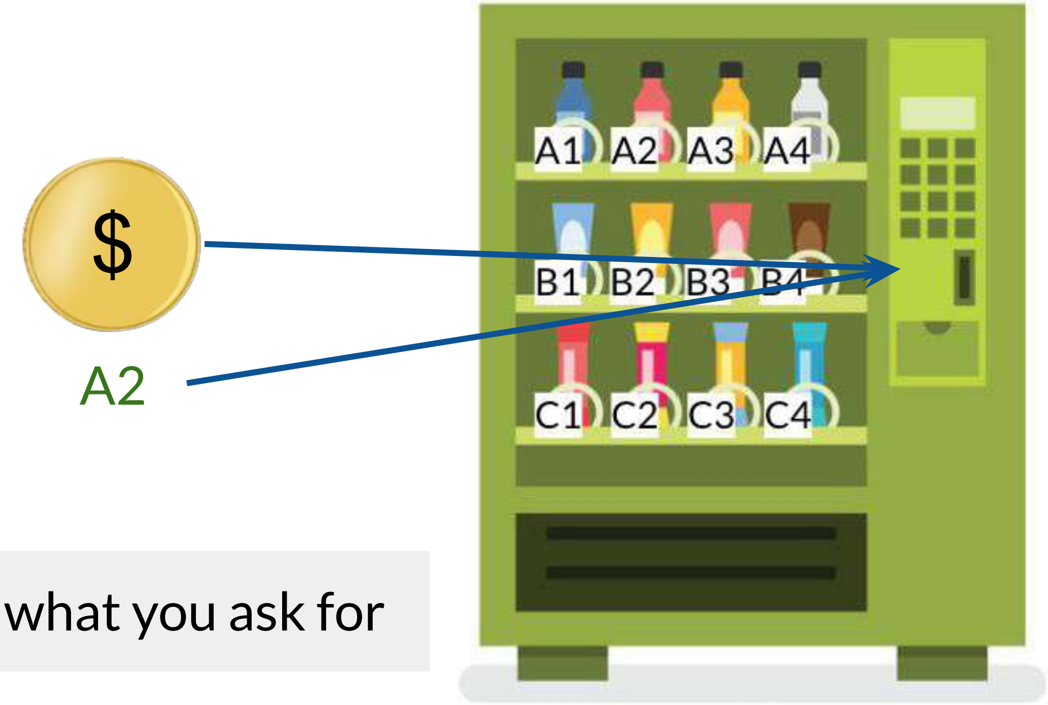
Conditional Generation

You get what you ask for

Conditional Generation

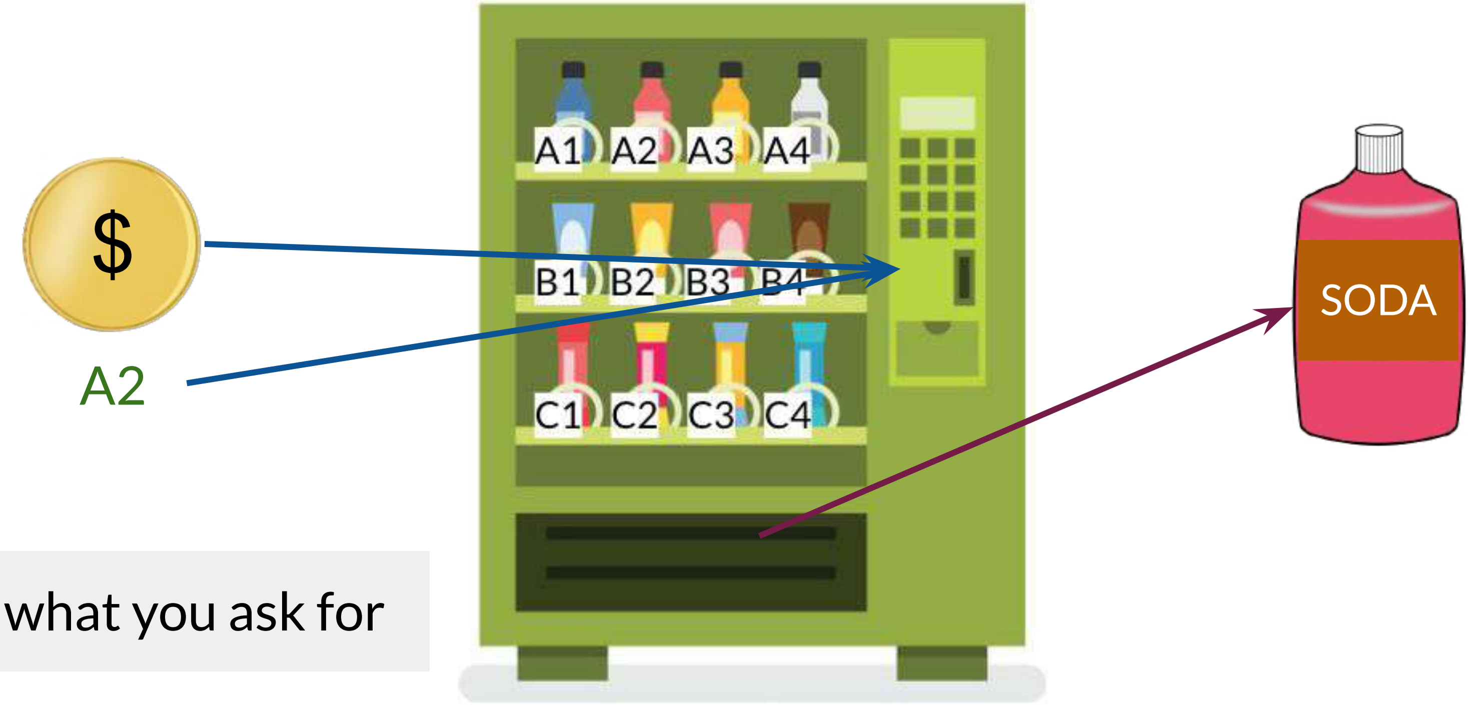
You get what you ask for

Conditional Generation



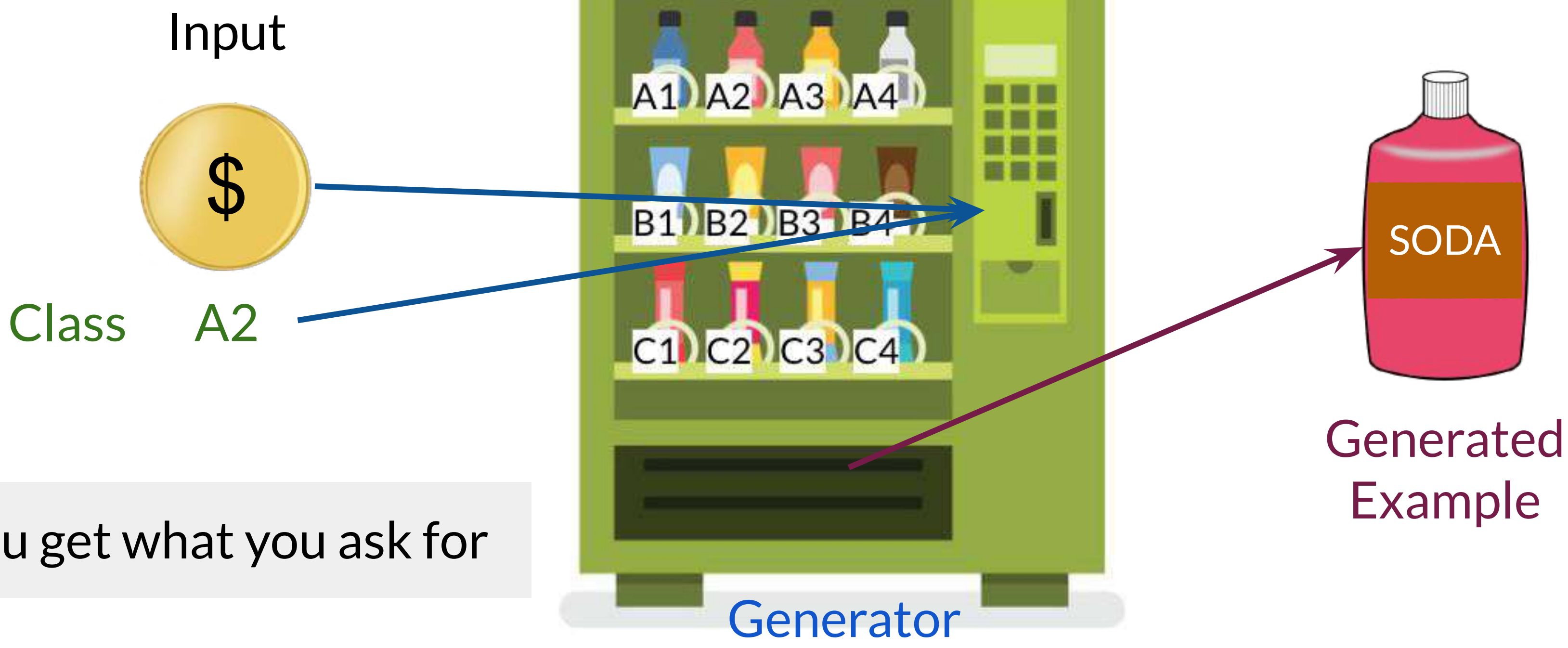
You get what you ask for

Conditional Generation



You get what you ask for

Conditional Generation



You get what you ask for

Conditional vs. Unconditional Generation

Conditional	Unconditional

Conditional vs. Unconditional Generation

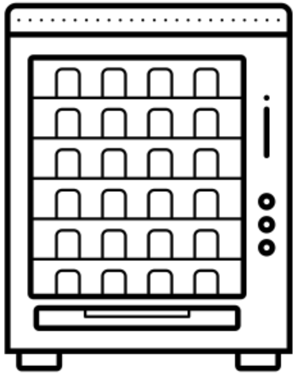
Conditional	Unconditional
Examples from **the classes you want**	Examples from *random classes*

Conditional vs. Unconditional Generation

Conditional	Unconditional
Examples from **the classes you want**	Examples from *random classes*
Training dataset needs to be **labeled**	Training dataset *doesn't need to be labeled*

Summary

- Conditional generation requires labeled datasets
- Examples can be generated for the selected class

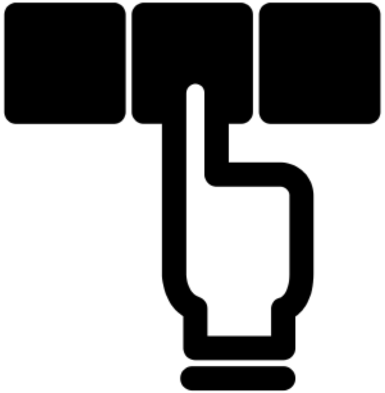


deeplearning.ai

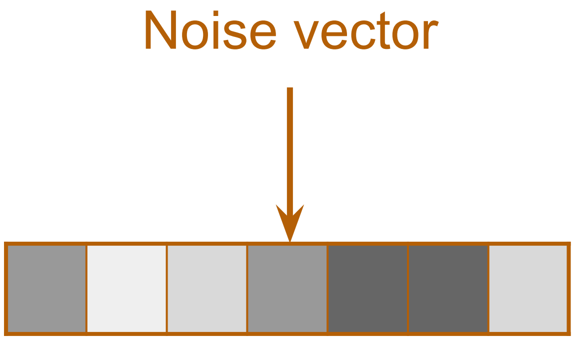
Conditional Generation: Inputs

Outline

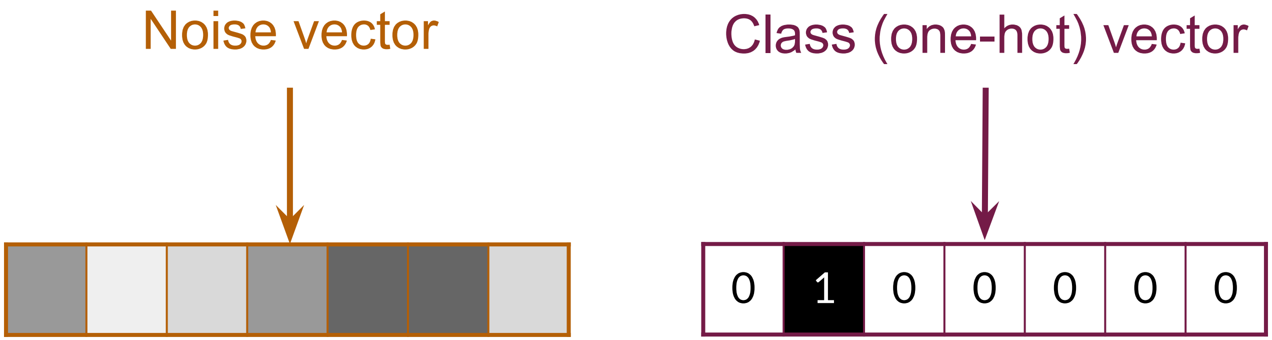
- How to tell the generator what type of example to produce
- Input representation for the discriminator



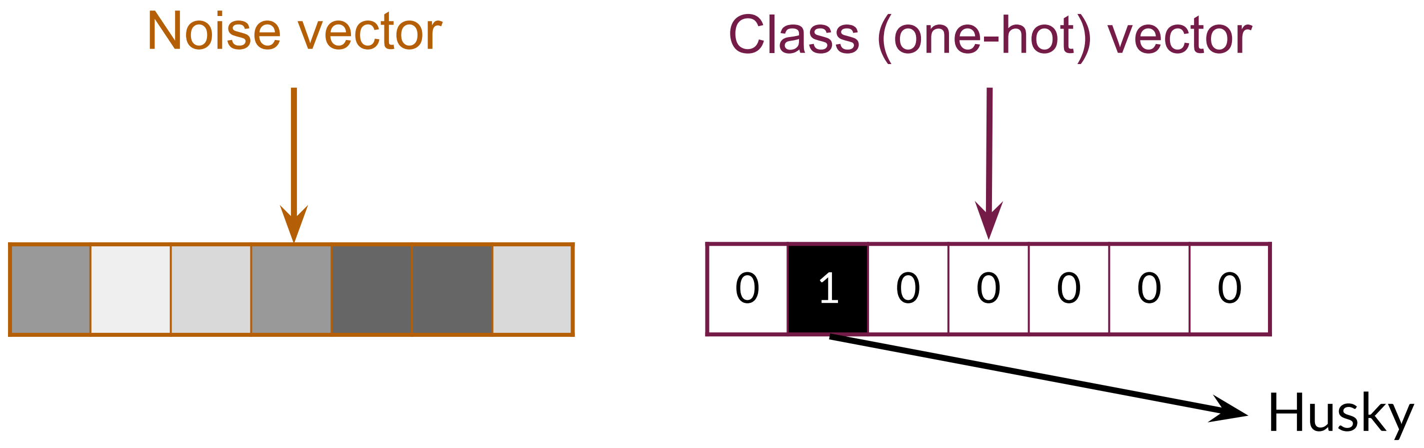
Generator Input



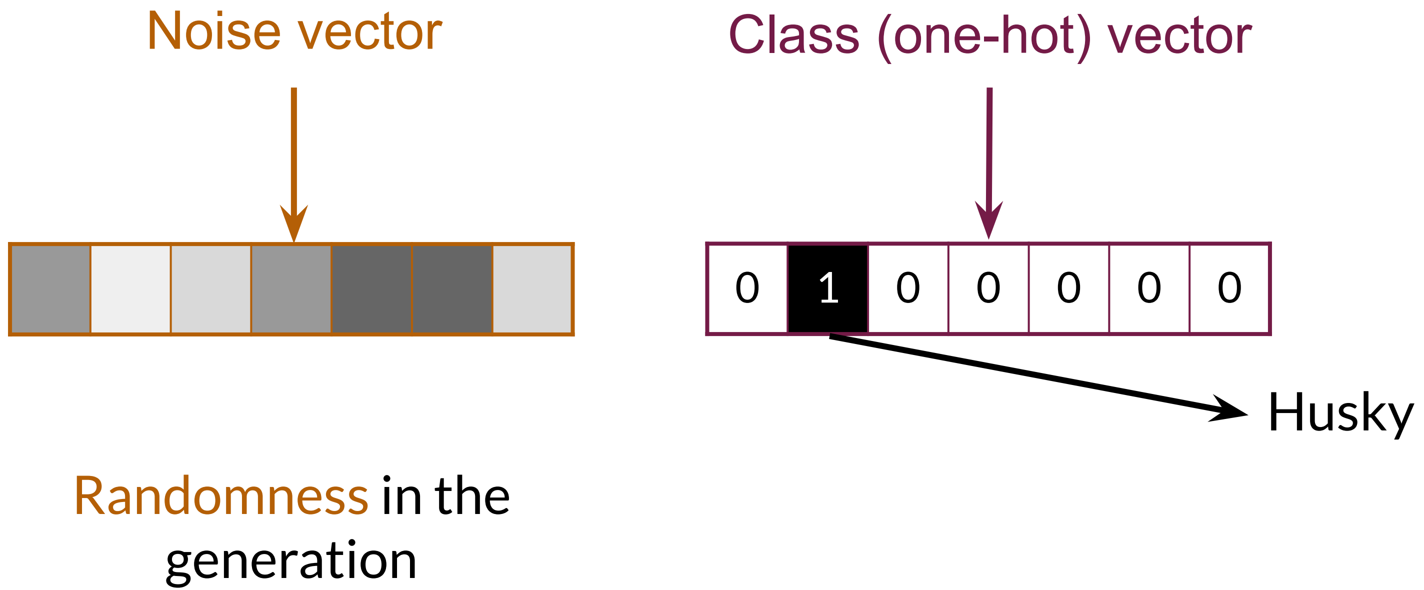
Generator Input



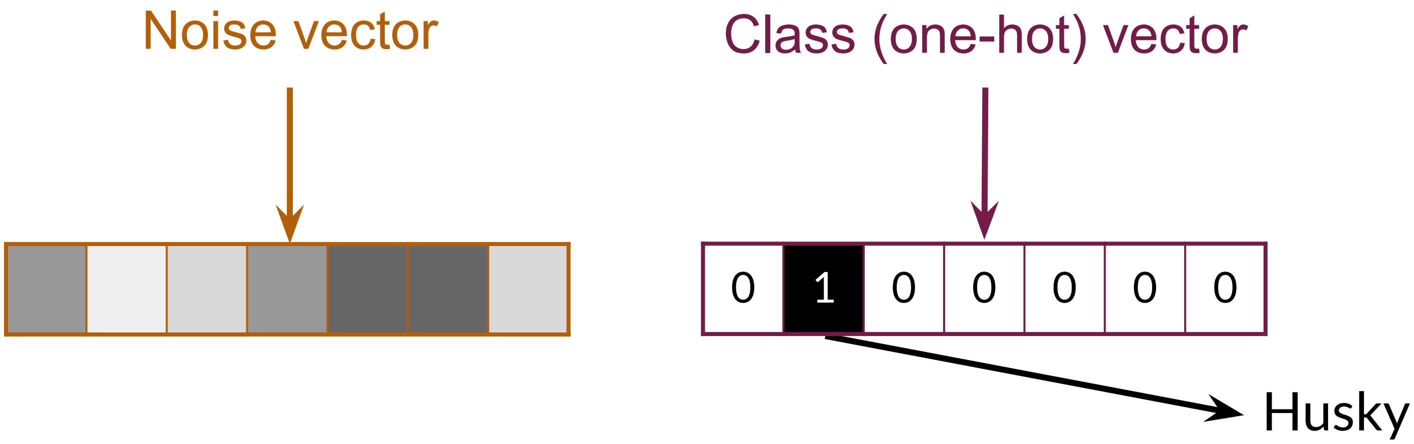
Generator Input



Generator Input



Generator Input



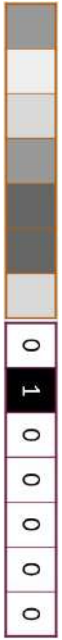
Randomness in the generation

Control in the generation

Generator Input

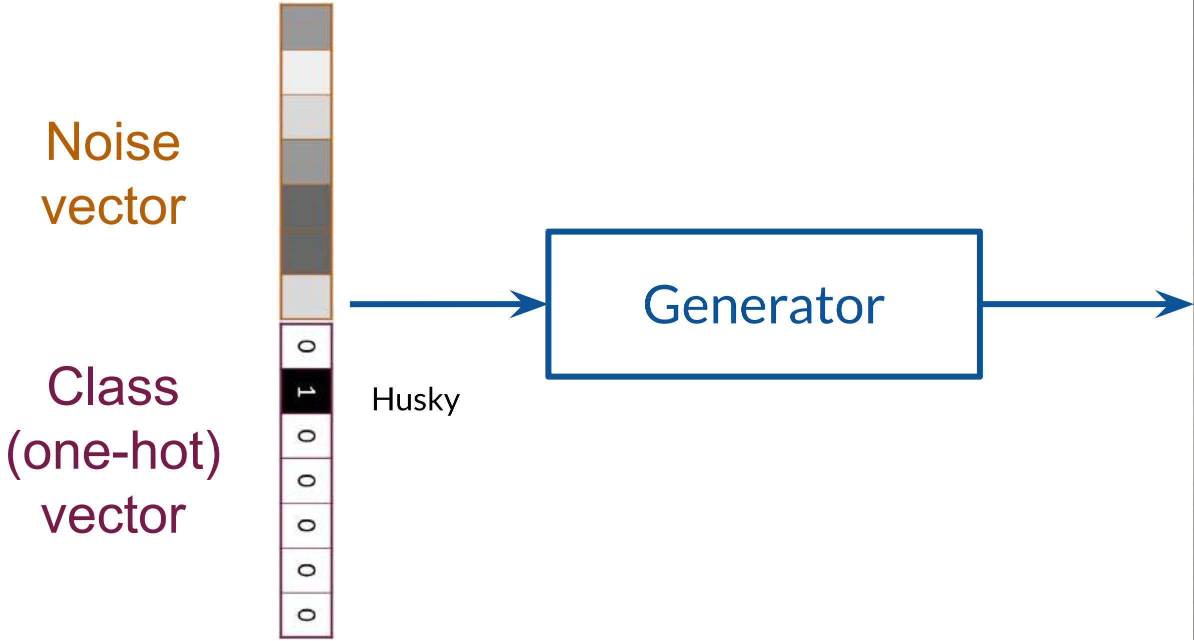
Noise vector

Class (one-hot) vector



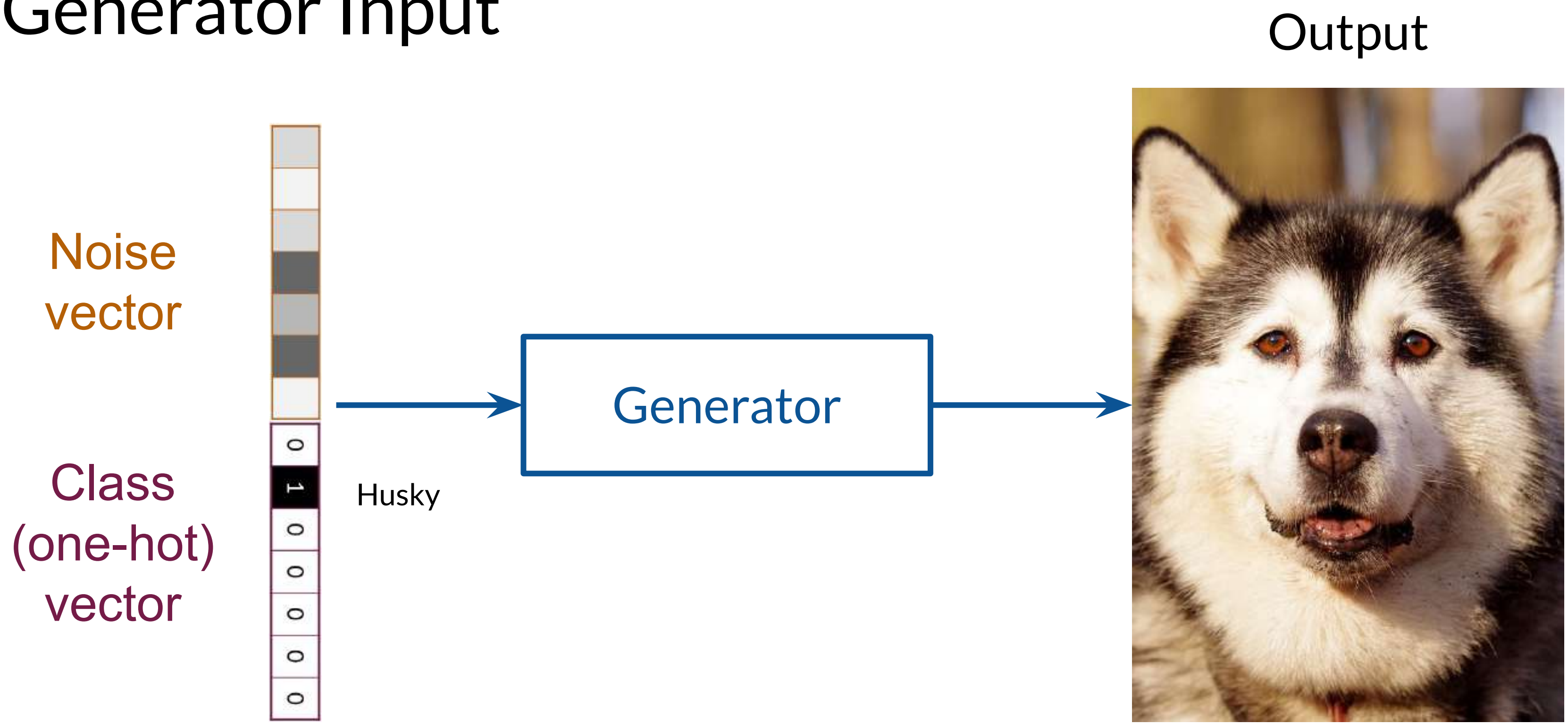
Husky

Generator Input



Output

Generator Input



Discriminator Input

Discriminator

Discriminator Input

Golden Retriever

Discriminator

Discriminator Input

Golden Retriever

Discriminator

Output

5%
Real

Discriminator Input

Discriminator

Golden Retriever

Discriminator Input

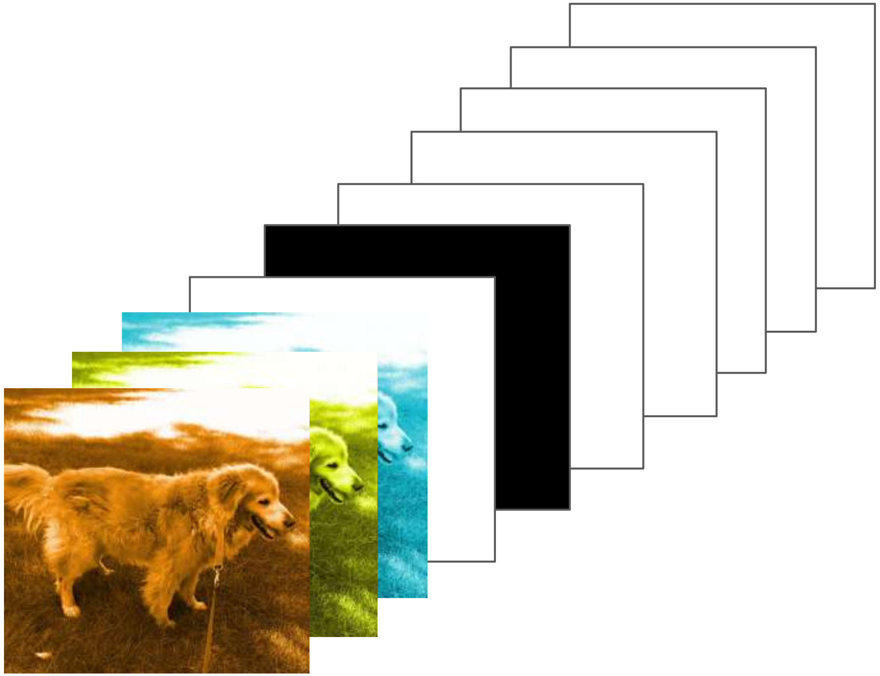
Golden Retriever

Discriminator

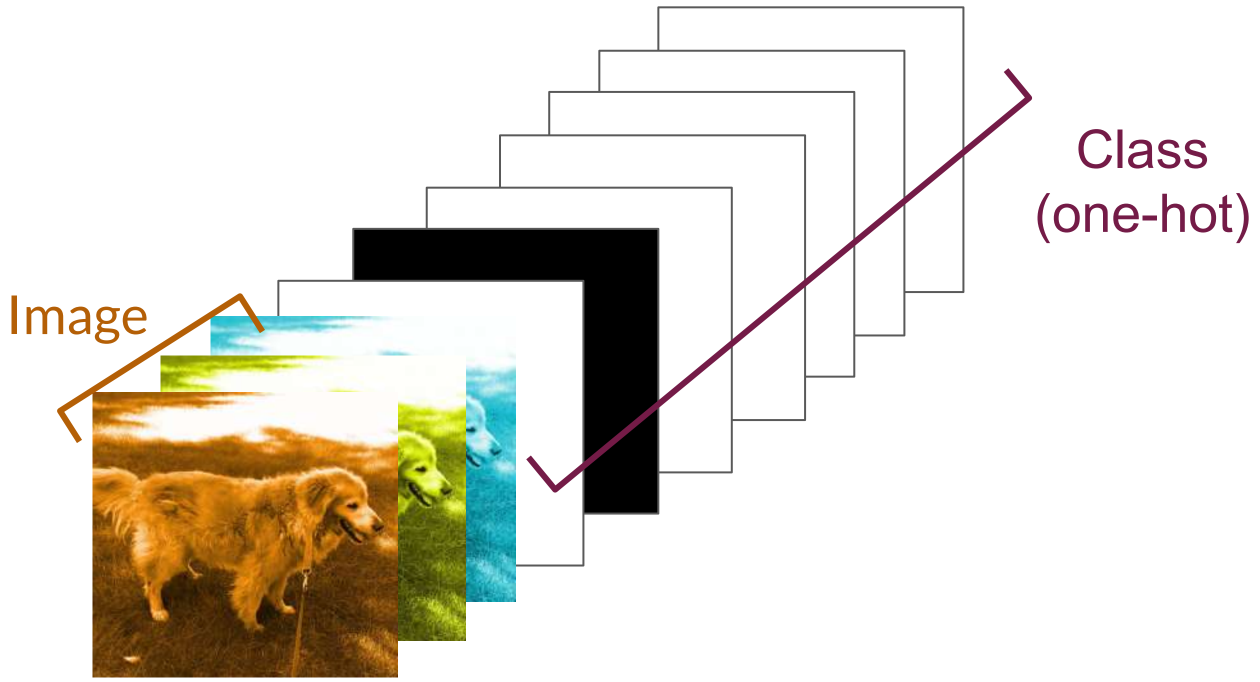
Output

98%
Real

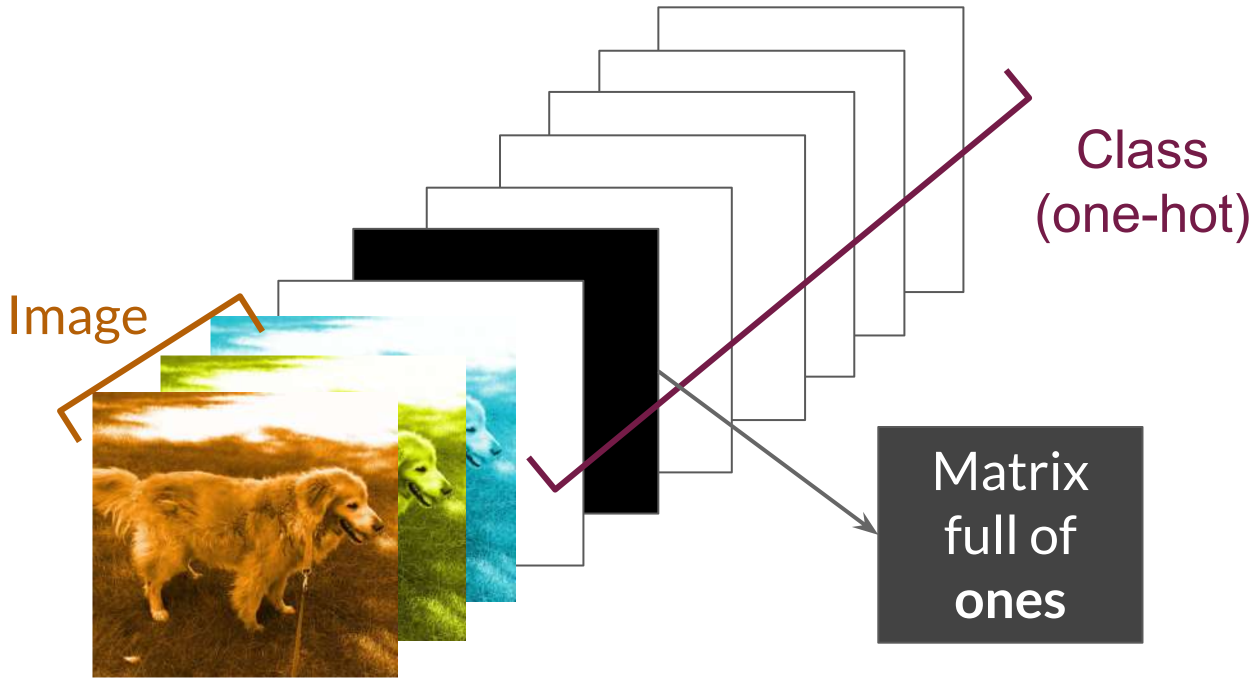
Discriminator Input



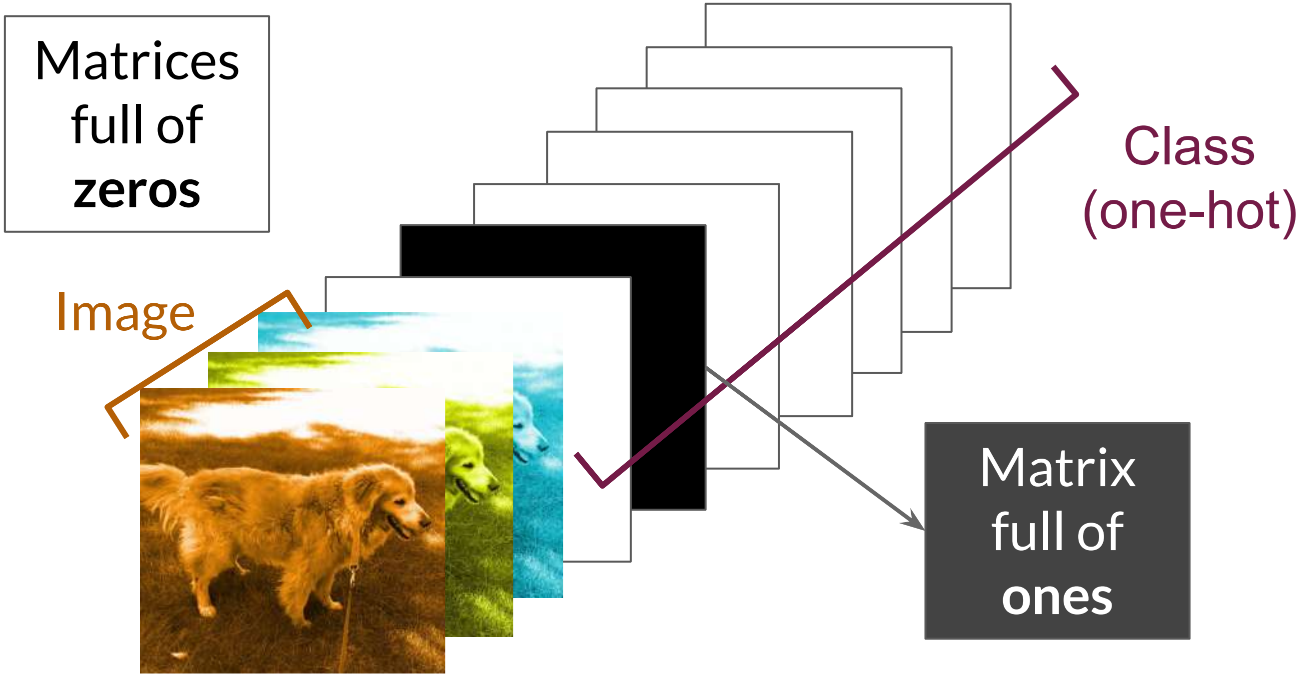
Discriminator Input



Discriminator Input



Discriminator Input



Summary

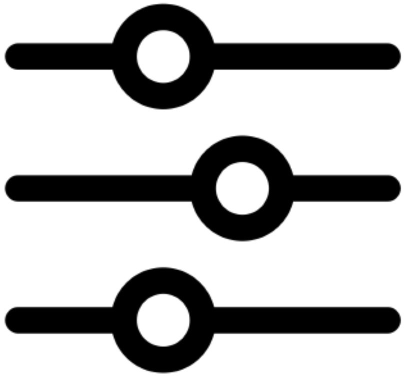
- The class is passed to the generator as one-hot vectors
- The class is passed to the discriminator as one-hot matrices
- The size of the vector and the number of matrices represent the number of classes

deeplearning.ai

Controllable Generation

Outline

- What is controllable generation
- How it compares to conditional generation



Controllable Generation

Change specific features of the output

Controllable Generation

Original Age Eyeglasses Gender Pose

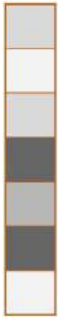
Change specific features of the output

Available from: https://arxiv.org/abs/1907.10786

Controllable Generation

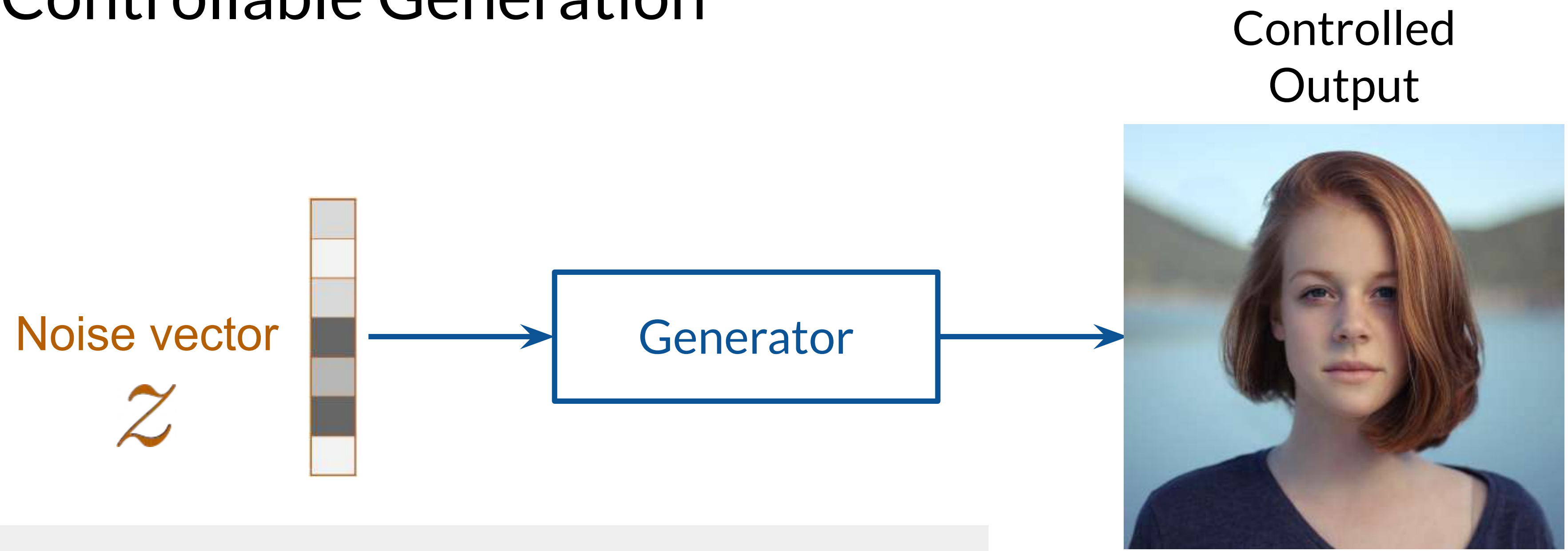
Noise vector

z



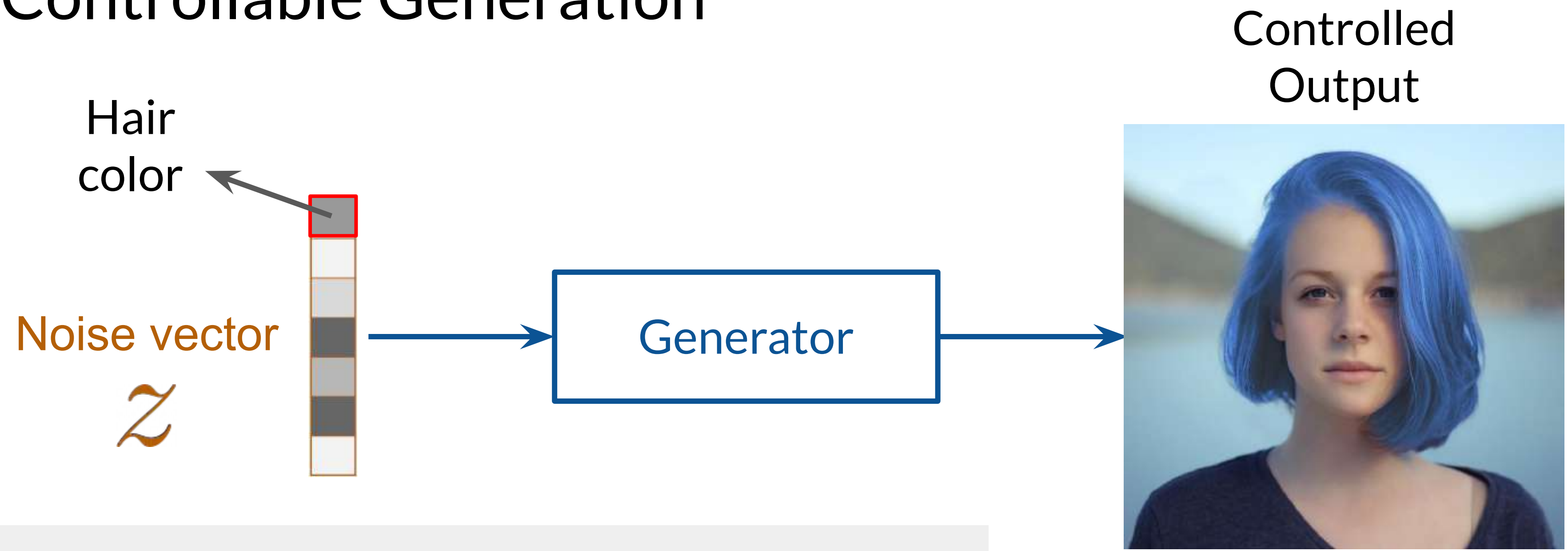
Tweak the input noise vector to get different features on the output

Controllable Generation



Tweak the input noise vector to get different features on the output

Controllable Generation



Tweak the input noise vector to get different features on the output

Controllable Generation vs. Conditional Generation

Controllable	Conditional

Controllable Generation vs. Conditional Generation

Controllable	Conditional
Examples with the **features that you want**	Examples from *the classes you want*

Controllable Generation vs. Conditional Generation

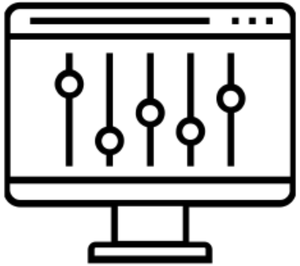
Controllable	Conditional
Examples with the **features that you want**	Examples from *the classes you want*
Training dataset **doesn't need to be labeled**	Training dataset *needs to be labeled*

Controllable Generation vs. Conditional Generation

Controllable	Conditional
Examples with the **features that you want**	Examples from *the classes you want*
Training dataset **doesn't need to be labeled**	Training dataset *needs to be labeled*
Manipulate the z vector input	*Append a class vector* to the input

Summary

- Controllable generation lets you control the features of the generated outputs
- It does not need a labeled training dataset
- The input vector is tweaked to get different features on the output

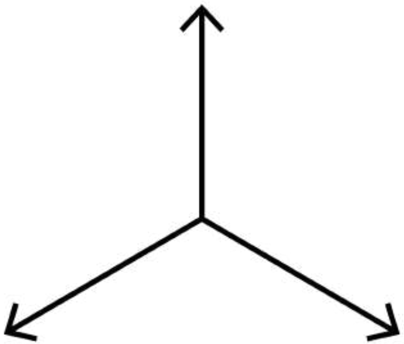


deeplearning.ai

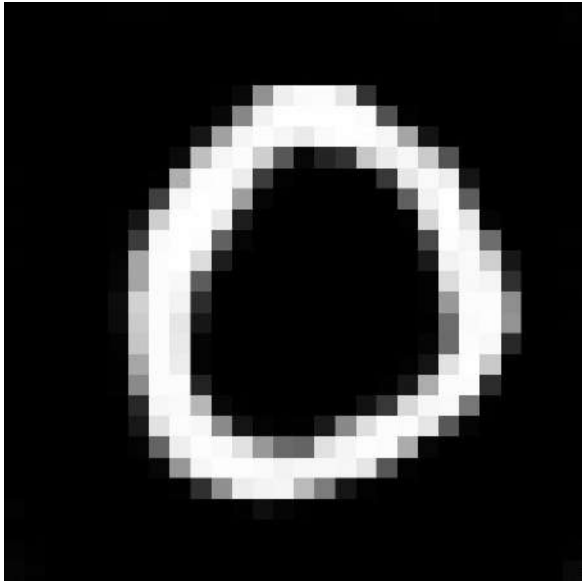
Vector Algebra in the Z-Space

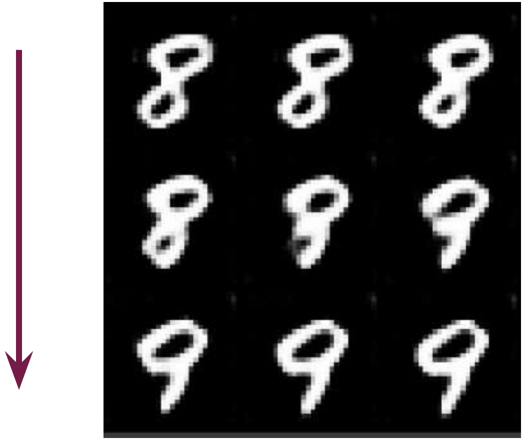
Outline

- Interpolation in the *Z*-space
- Modifying the noise vector *z* to control desired features



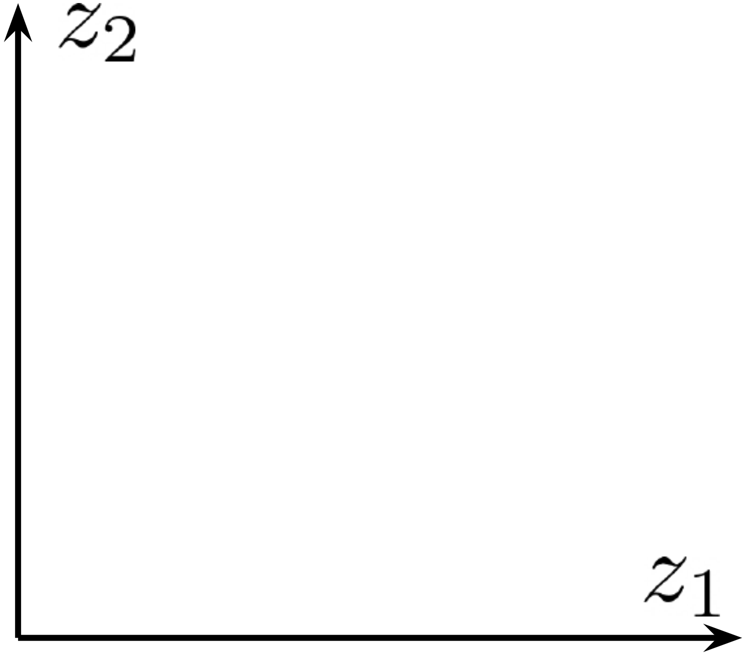
Interpolation Using the Z-Space





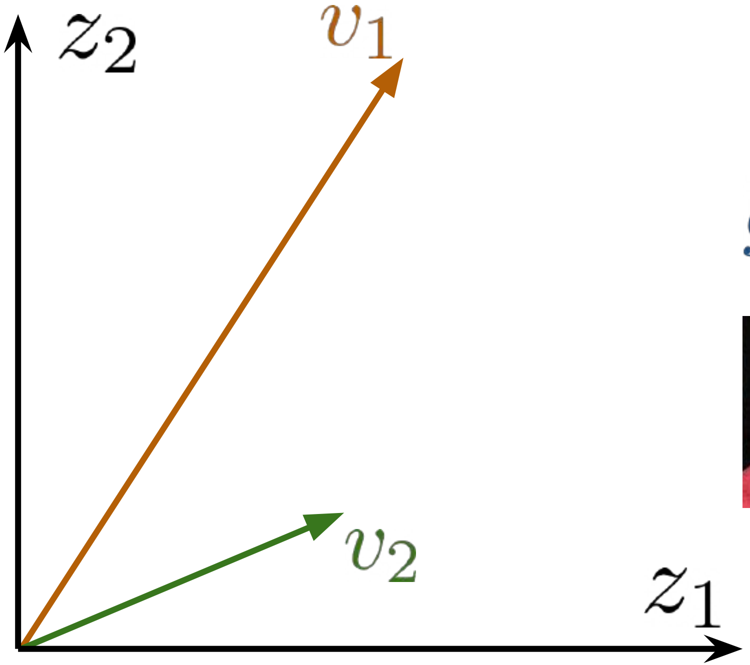
How an image morphs into another

Interpolation Using the *Z*-Space



Z-Space with Noise Vectors

Interpolation Using the Z-Space

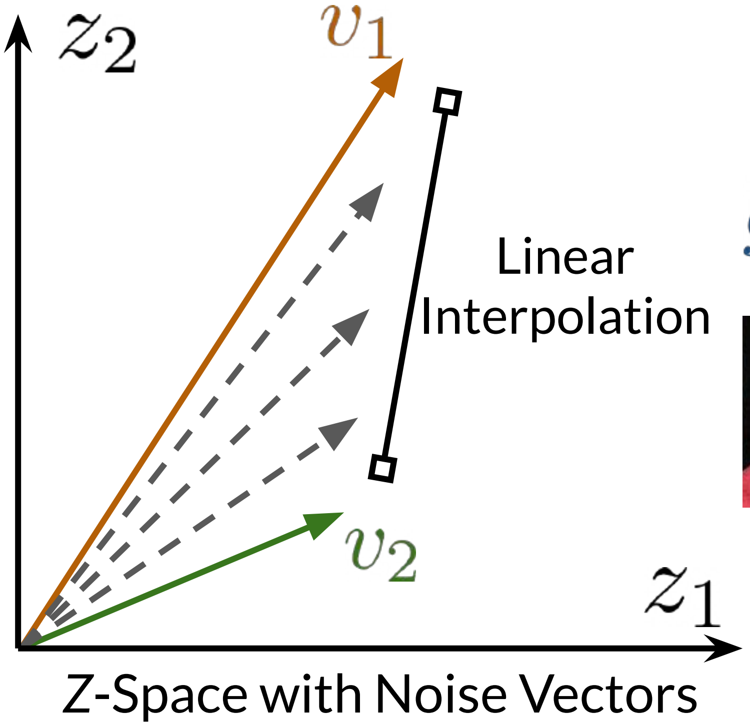


Z-Space with Noise Vectors

$g(v_1)$

$g(v_2)$

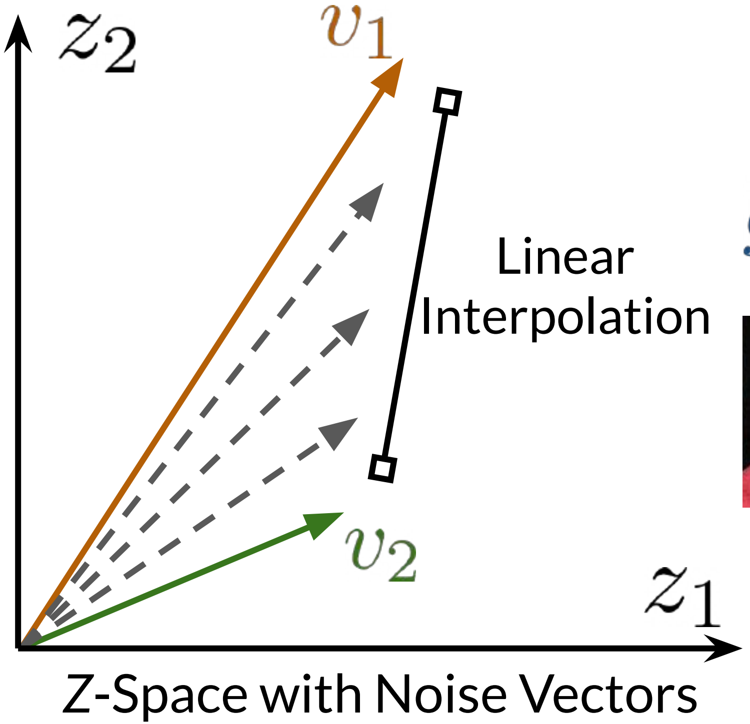
Interpolation Using the Z-Space



$g(v_1)$

$g(v_2)$

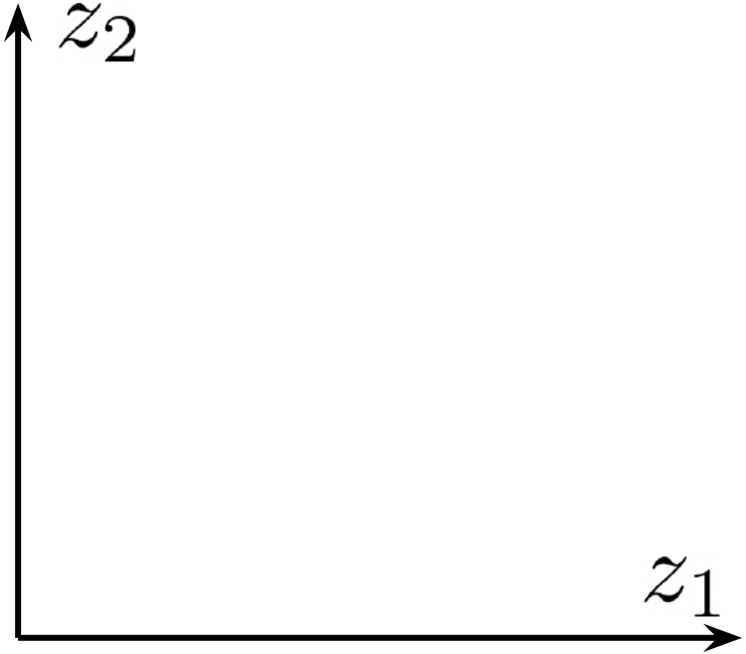
Interpolation Using the Z-Space



Intermediate images using the Z-space

$g(v_1)$ ⟷ $g(v_2)$

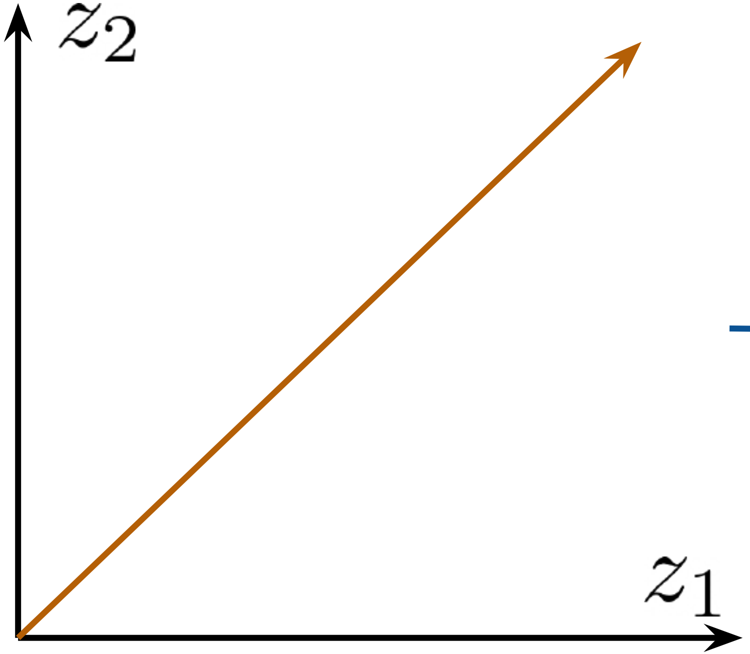
Z-Space and Controllable Generation



Z-Space with Noise Vectors

Generator

Z-Space and Controllable Generation

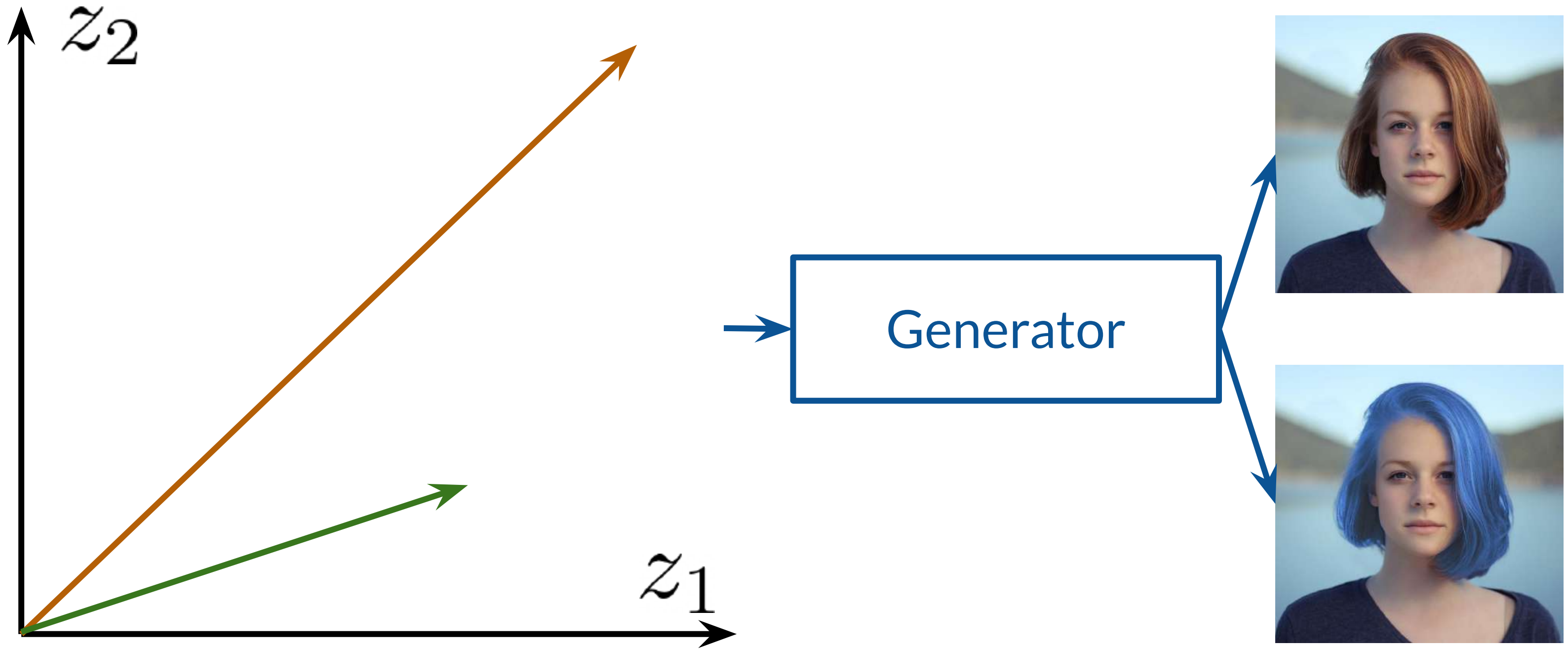


Z-Space with Noise Vectors

Generator

Output

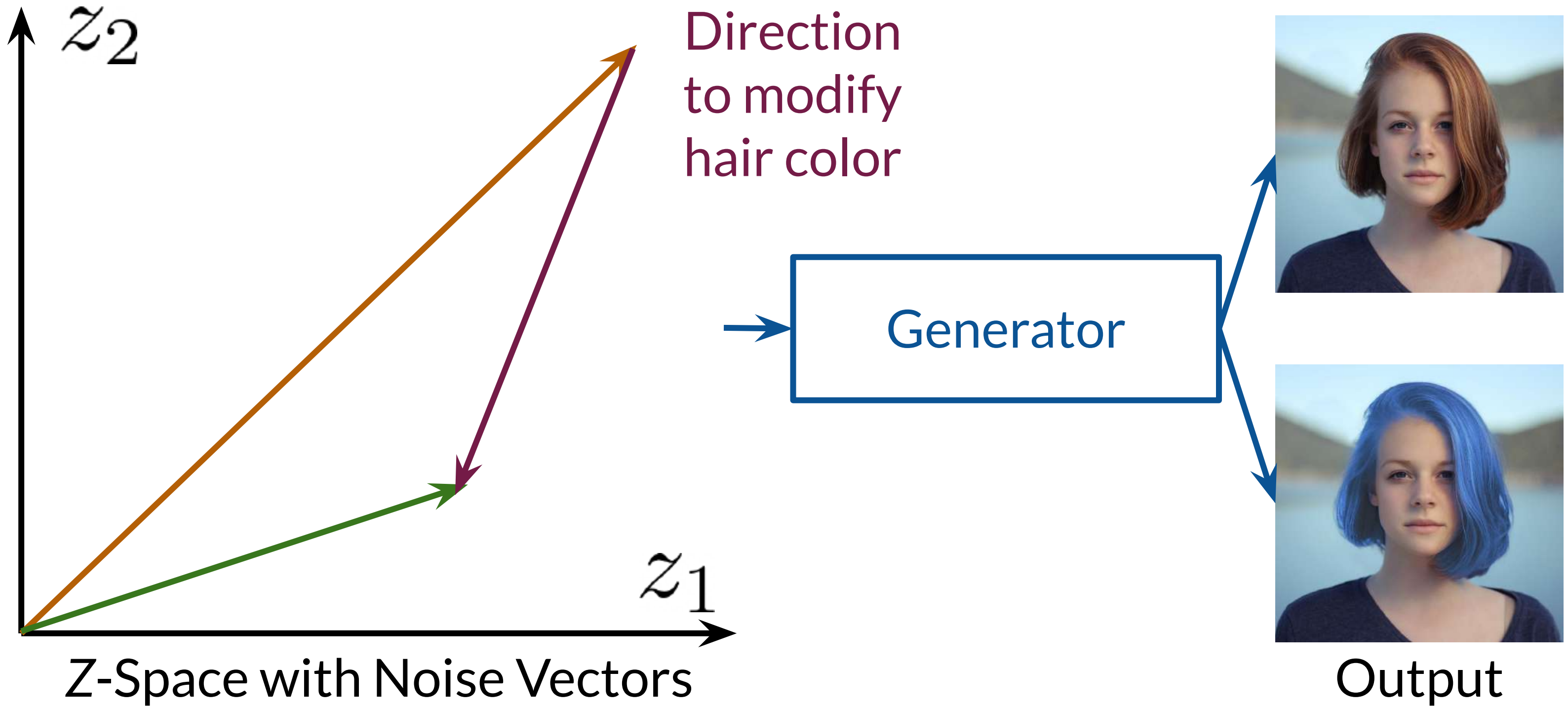
Z-Space and Controllable Generation



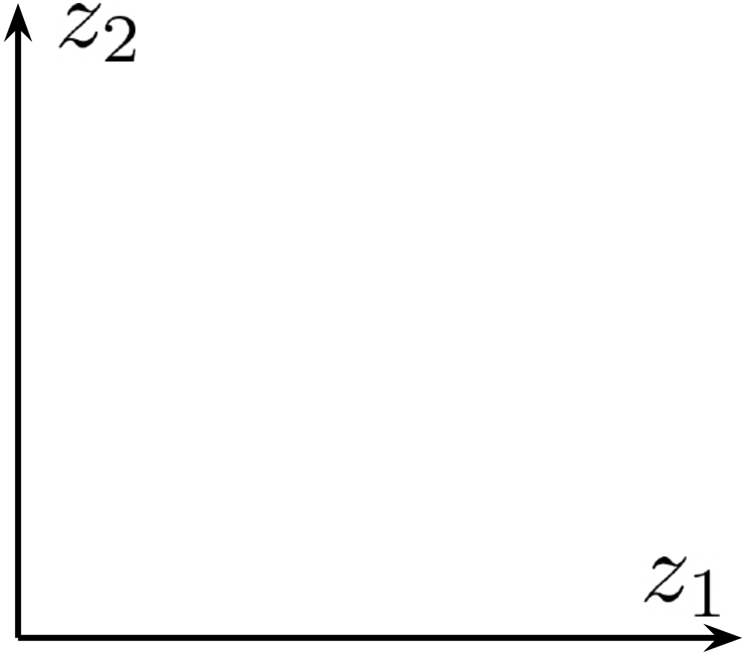
Z-Space with Noise Vectors

Output

Z-Space and Controllable Generation

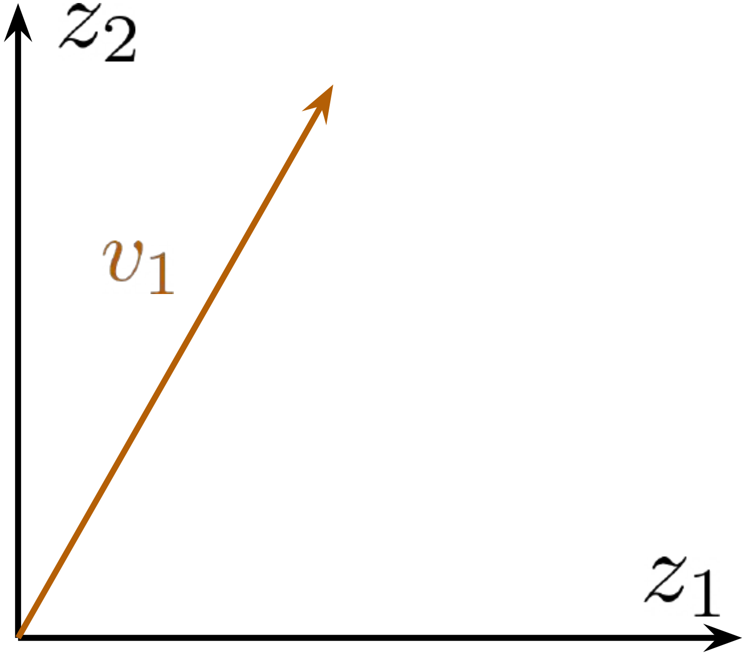


Z-Space and Controllable Generation



Z-Space with Noise Vectors

Z-Space and Controllable Generation

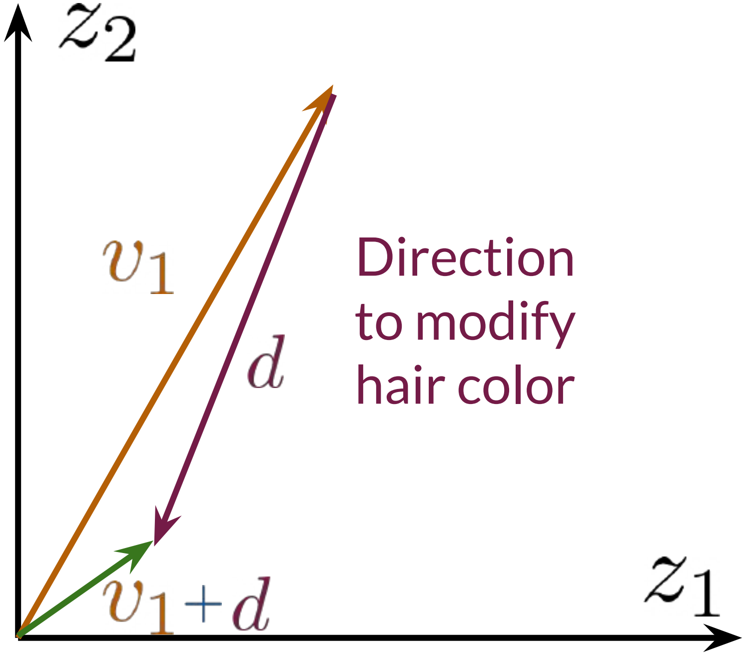


Z-Space with Noise Vectors

Original output

$g(v_1)$

Z-Space and Controllable Generation

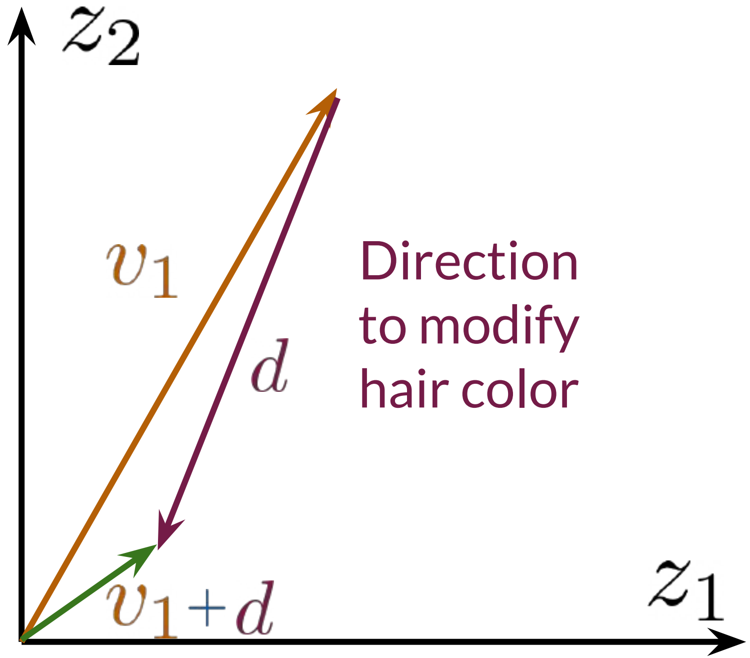


Z-Space with Noise Vectors

Original output

$g(v_1) \rightarrow$

Z-Space and Controllable Generation



Z-Space with Noise Vectors

Original output

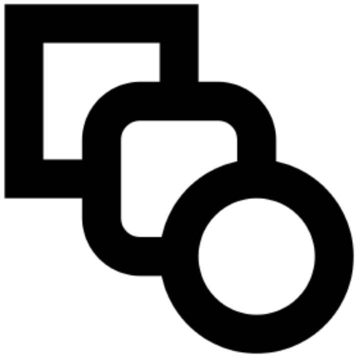
$g(v_1)$

Controlled output

$g(v_1+d)$

Summary

- To control output features, you need to find directions in the Z-space
- To modify your output, you move around in the Z-space



deeplearning.ai

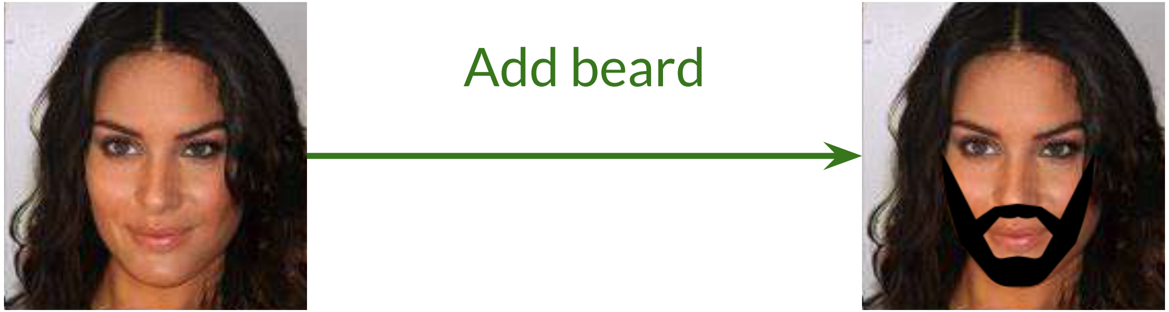
Challenges with Controllable Generation

Outline

- Output feature correlation
- Z-space entanglement

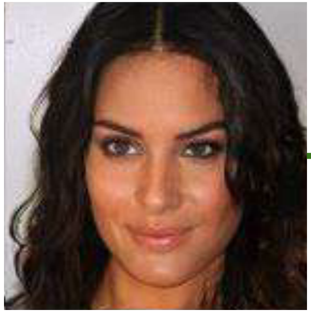
Feature Correlation

Uncorrelated
Features



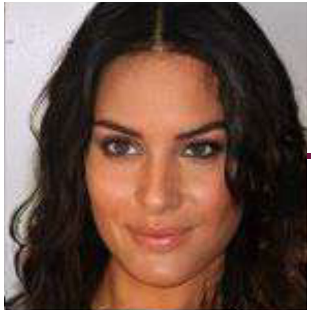
Feature Correlation

Uncorrelated Features



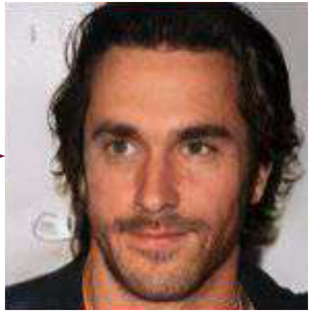
Add beard

Correlated Features



Add beard

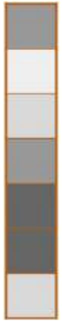
Make more masculine



Z-Space Entanglement

Noise vector

z



Glasses

Beard

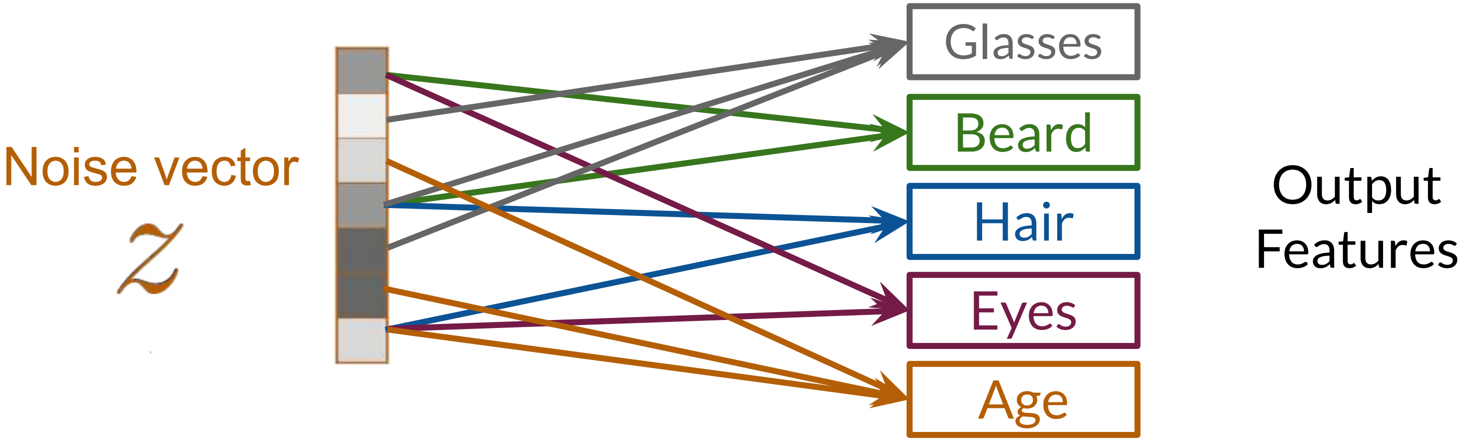
Hair

Eyes

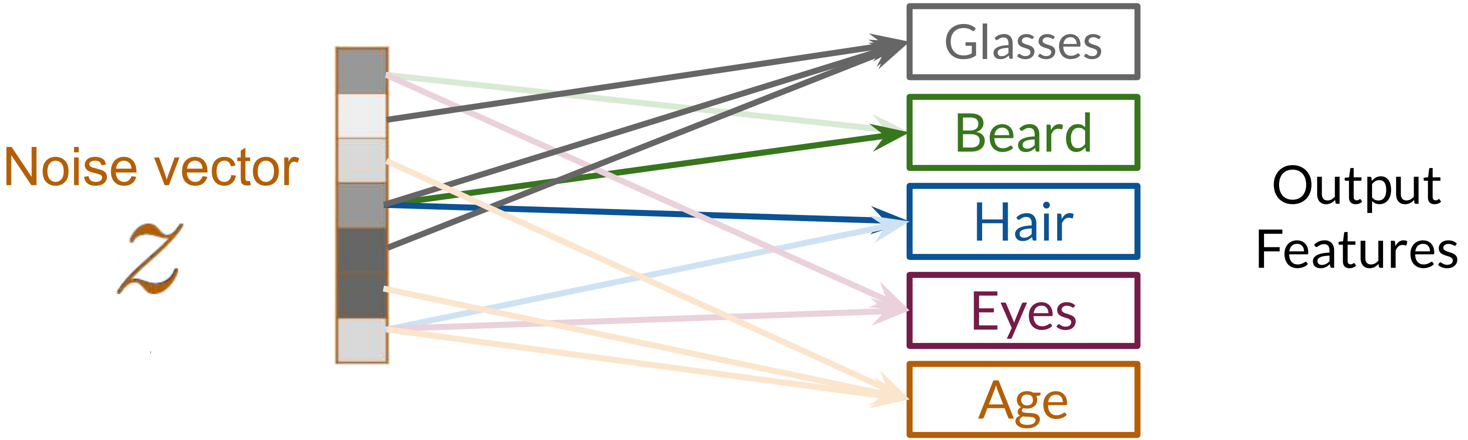
Age

Output Features

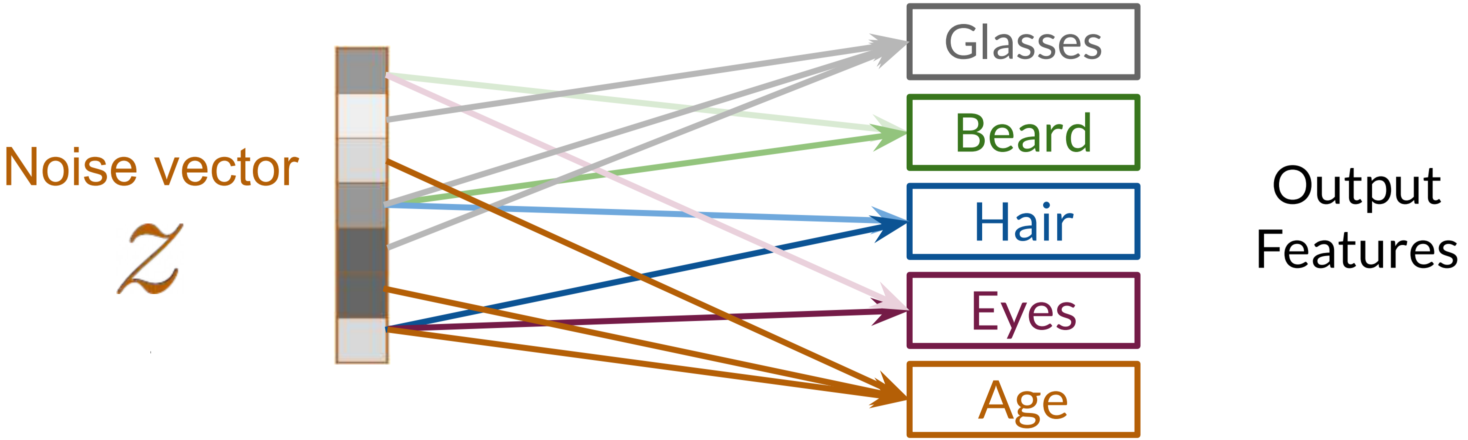
Z-Space Entanglement



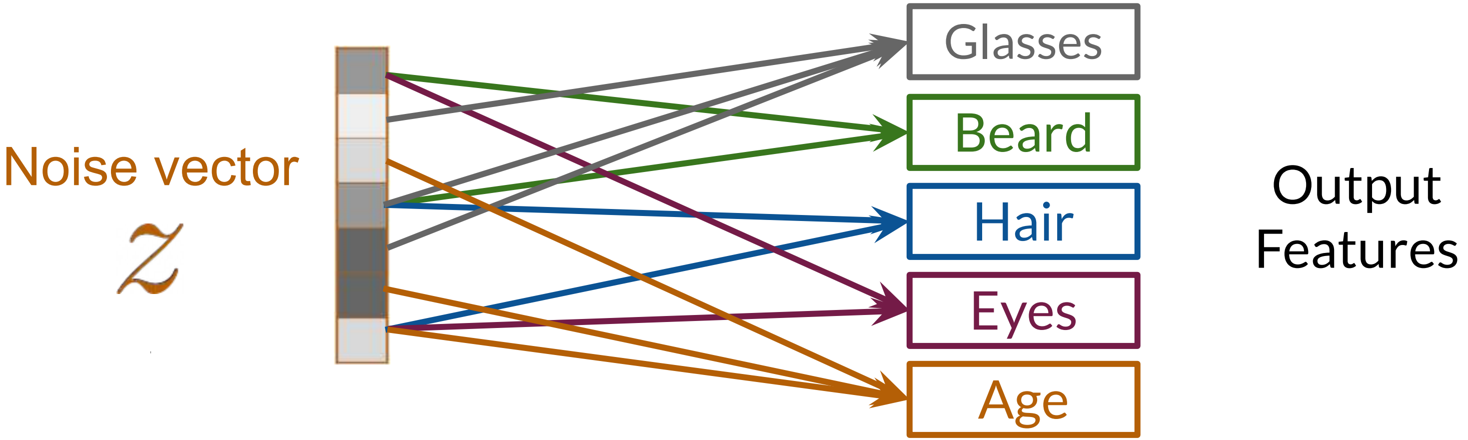
Z-Space Entanglement



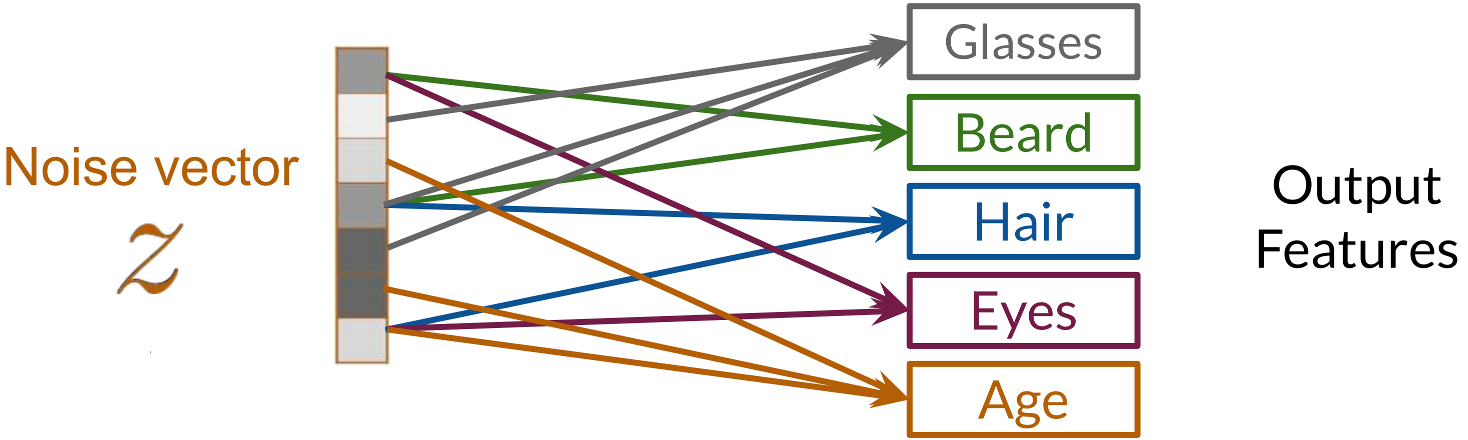
Z-Space Entanglement



Z-Space Entanglement

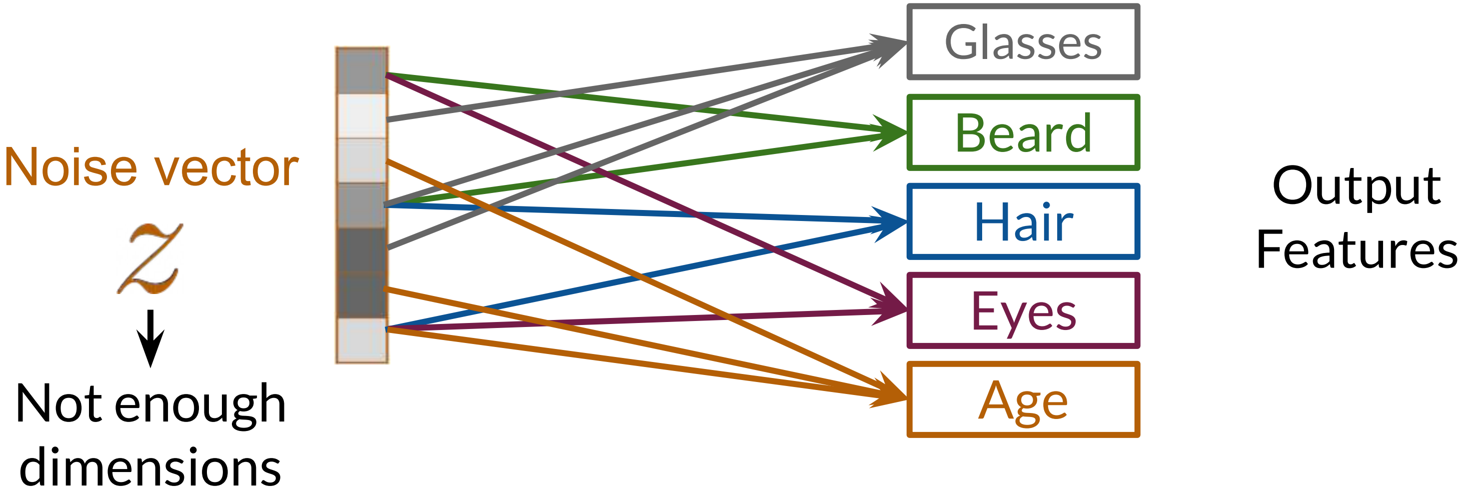


Z-Space Entanglement



It is not possible to control single output features

Z-Space Entanglement



It is not possible to control single output features

Summary

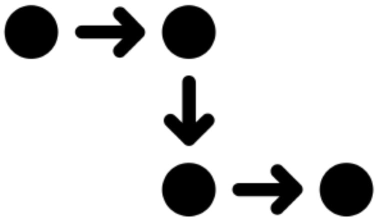
- When trying to control one feature, others that are correlated change
- *Z*-space entanglement makes controllability difficult, if not impossible
- Entanglement happens when z does not have enough dimensions

Classifier Gradients

deeplearning.ai

Outline

- How to use classifiers to find directions in the Z-space
- Requirements to use this method



Classifier Gradients

Classifier Gradients

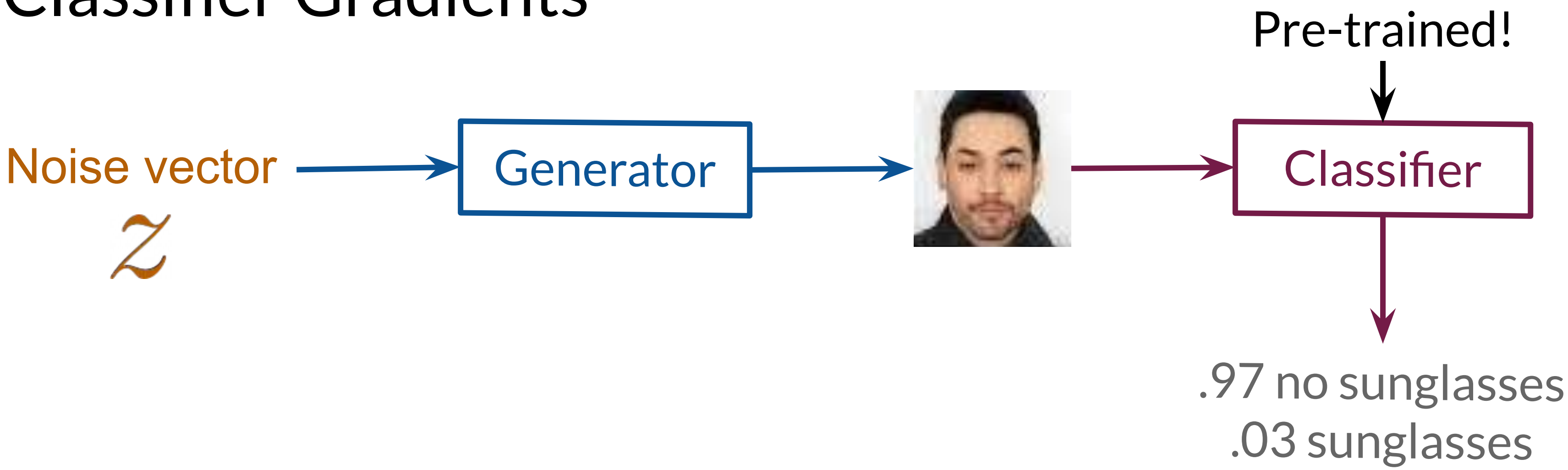
Noise vector

z

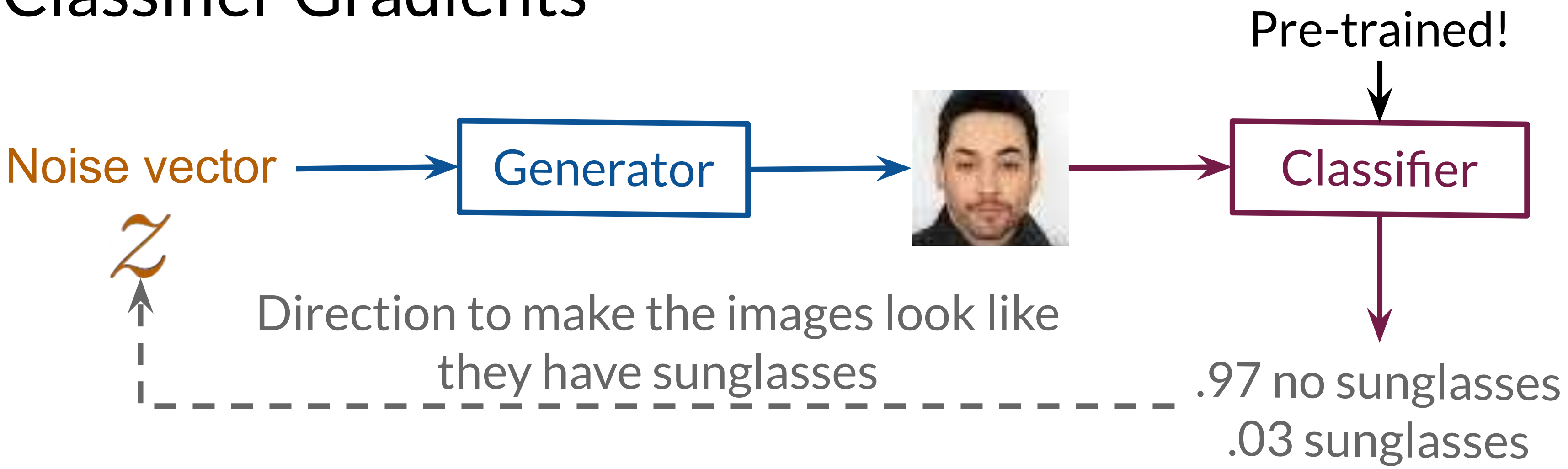
Classifier Gradients

Noise vector z → Generator → 

Classifier Gradients



Classifier Gradients



Classifier Gradients

Pre-trained!

Noise vector → Generator →  → Classifier

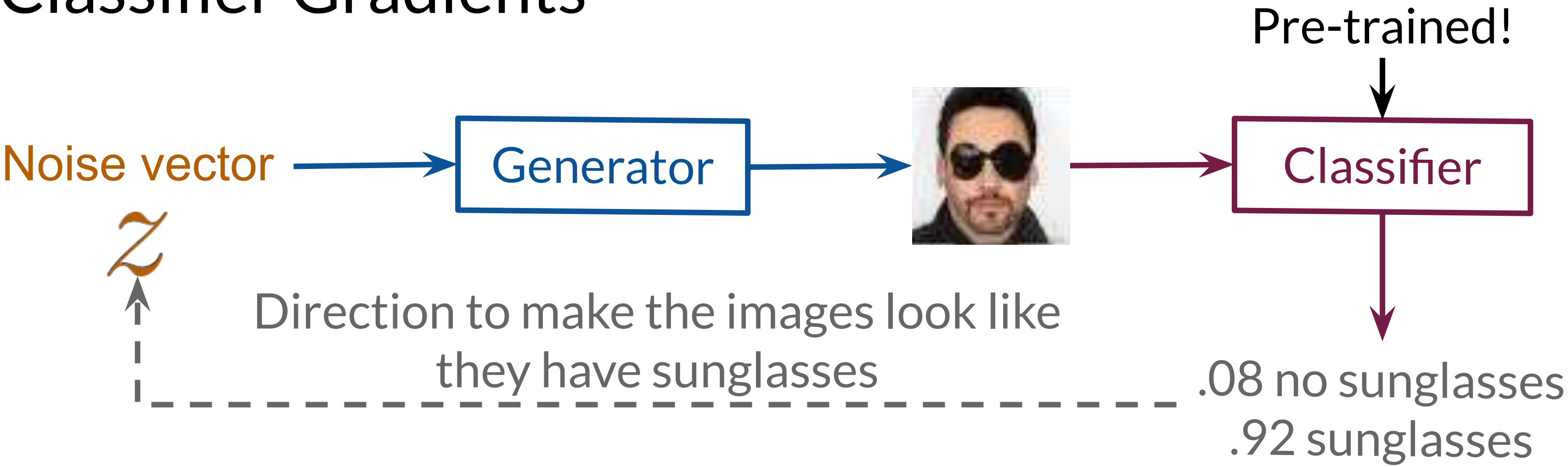
z

Direction to make the images look like they have sunglasses

.25 no sunglasses
.75 sunglasses

Modify **just** the **noise vector** until the feature emerges

Classifier Gradients



Modify **just** the noise vector until the feature emerges

Summary

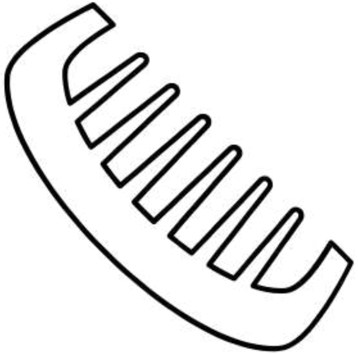
- Classifiers can be used to find directions in the Z-space
- To find directions, the updates are done just to the noise vector

deeplearning.ai

Disentanglement

Outline

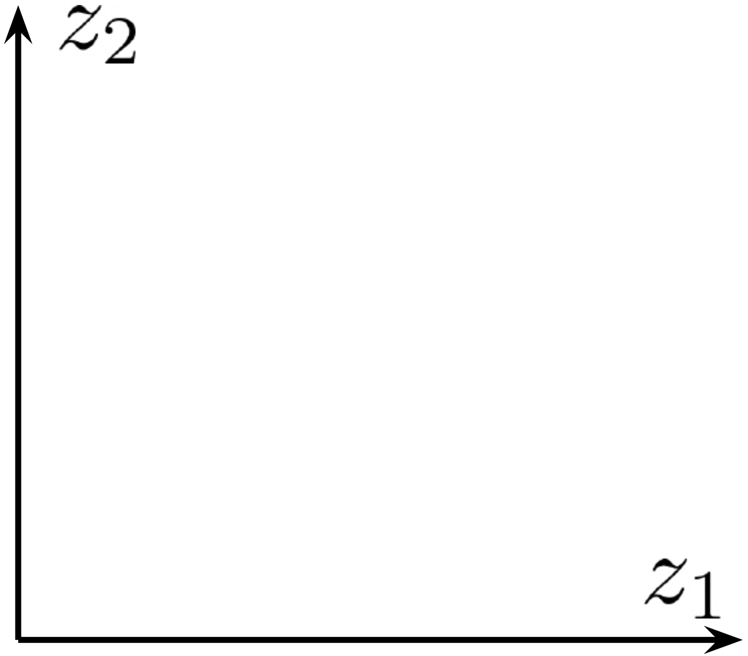
- What a disentangled Z-space means
- Ways to encourage disentangled Z-spaces



Disentangled Z-Space

$$v_1 = [\ 1,\ 2,\ 3,\ldots\ ]$$

$$v_2 = [\ 5,\ 6,\ 7,\ldots\ ]$$



Z-Space with Noise Vectors

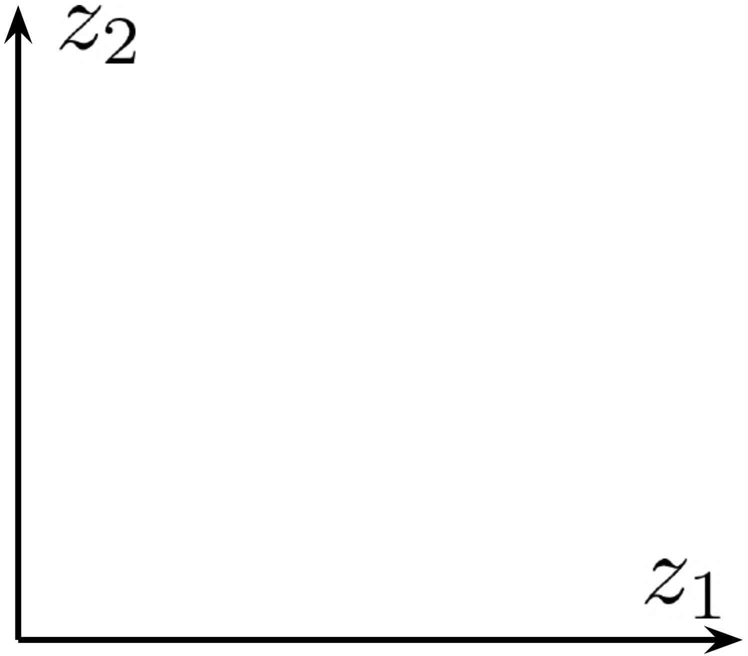
Disentangled Z-Space

z_1

$$v_1 = [\,1, 2, 3, \ldots\,]$$

$$v_2 = [\,5, 6, 7, \ldots\,]$$

Hair color



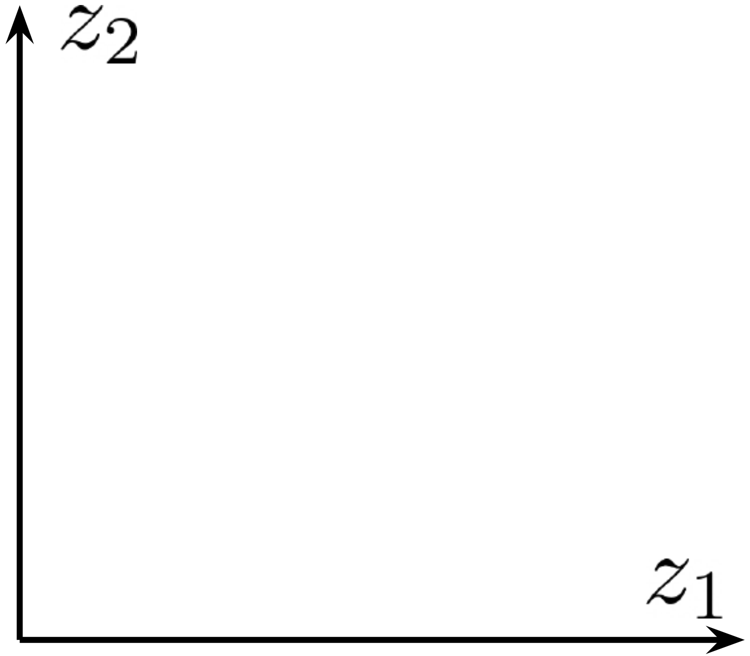
Z-Space with Noise Vectors

Disentangled Z-Space

$$\begin{array}{llll} & z_1 & z_2 & \\ v_1 = [& 1, & 2, & 3, \dots] \\ v_2 = [& 5, & 6, & 7, \dots] \end{array}$$

Hair color

Hair length



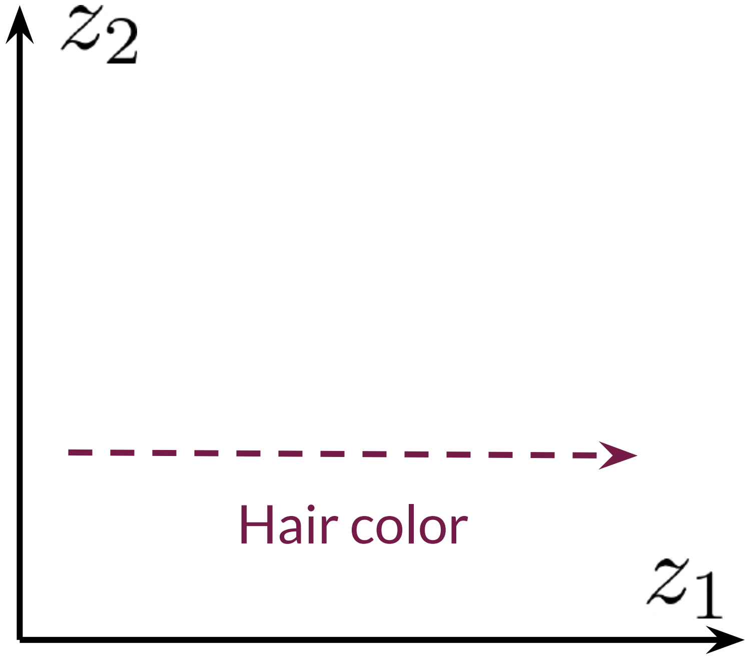
Z-Space with Noise Vectors

Disentangled Z-Space

$$\begin{array}{c|c|c|c} & z_1 & z_2 & \\ v_1 = [& 1, & 2, & 3, \ldots] \\ v_2 = [& 5, & 6, & 7, \ldots] \end{array}$$

Hair color

Hair length



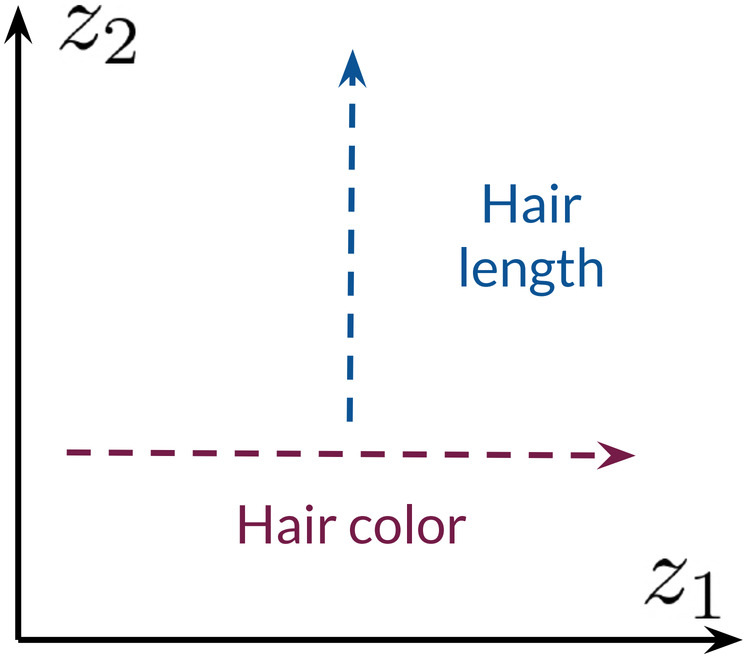
Z-Space with Noise Vectors

Disentangled Z-Space

$$\begin{array}{rcl} & & \quad\; z_1 \quad z_2 \\ v_1 & = & [\, 1,\; 2,\; 3, \ldots \,] \\ v_2 & = & [\, 5,\; 6,\; 7, \ldots \,] \end{array}$$

Hair color

Hair length



Z-Space with Noise Vectors

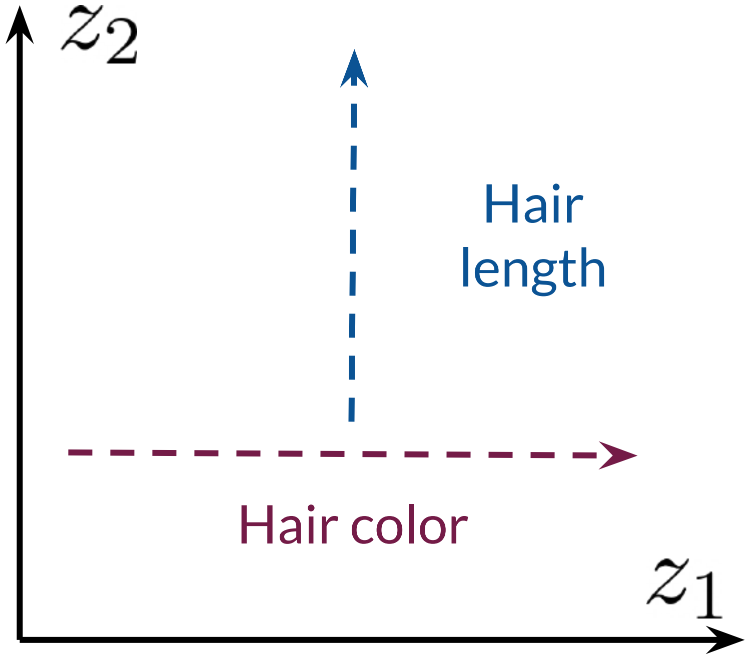
Disentangled Z-Space

$$\begin{array}{cccc} & z_1 & z_2 & \\ v_1 = [& 1, & 2, & 3, \ldots] \\ v_2 = [& 5, & 6, & 7, \ldots] \end{array}$$

Hair color

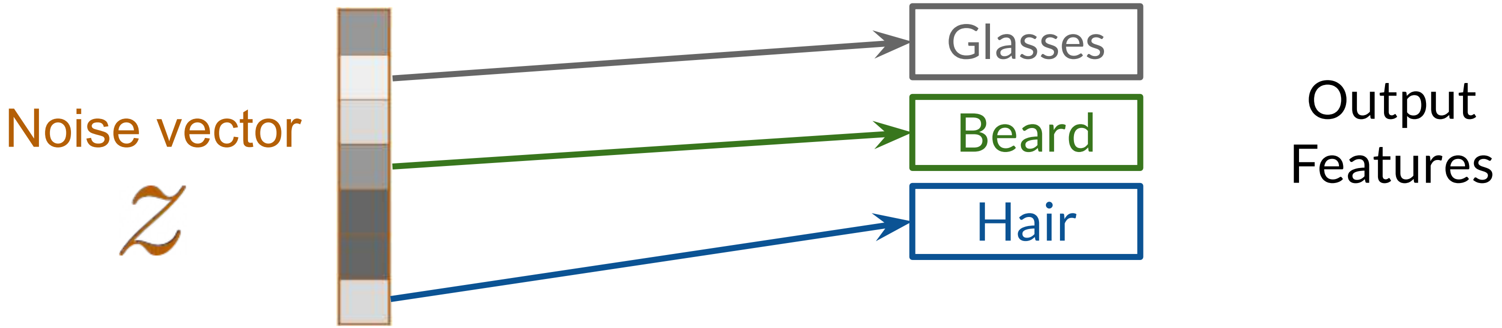
Hair length

Latent factors of variation

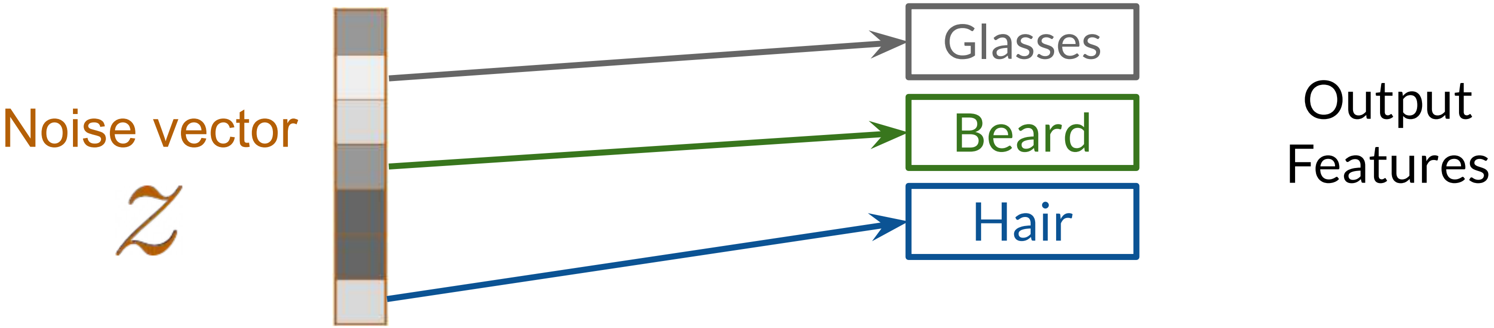


Z-Space with Noise Vectors

Disentangled Z-Space

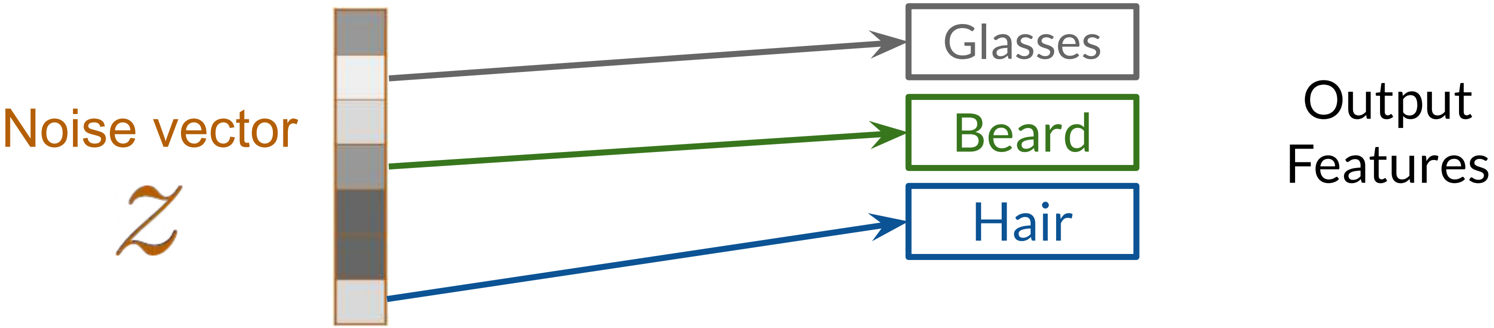


Disentangled Z-Space

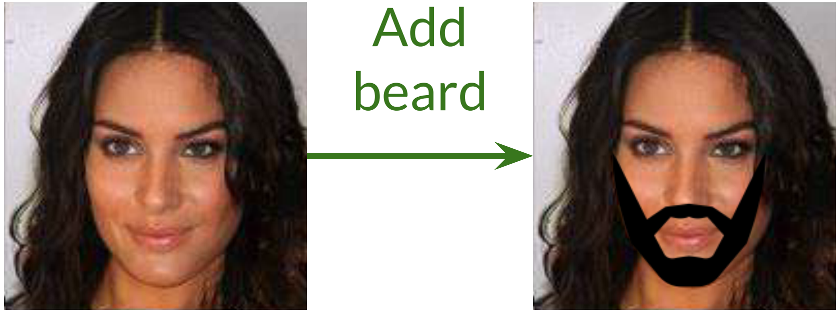


Changes to one feature don't affect the others

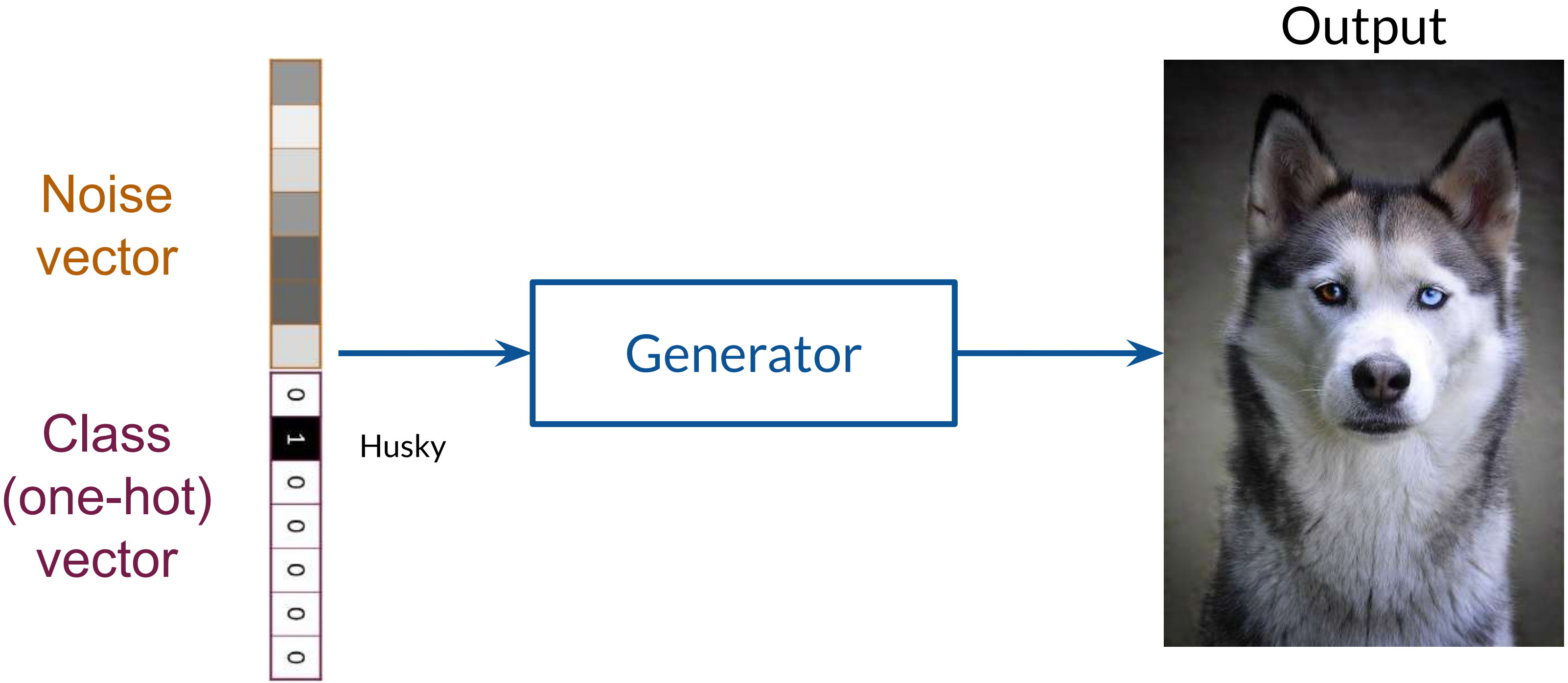
Disentangled Z-Space



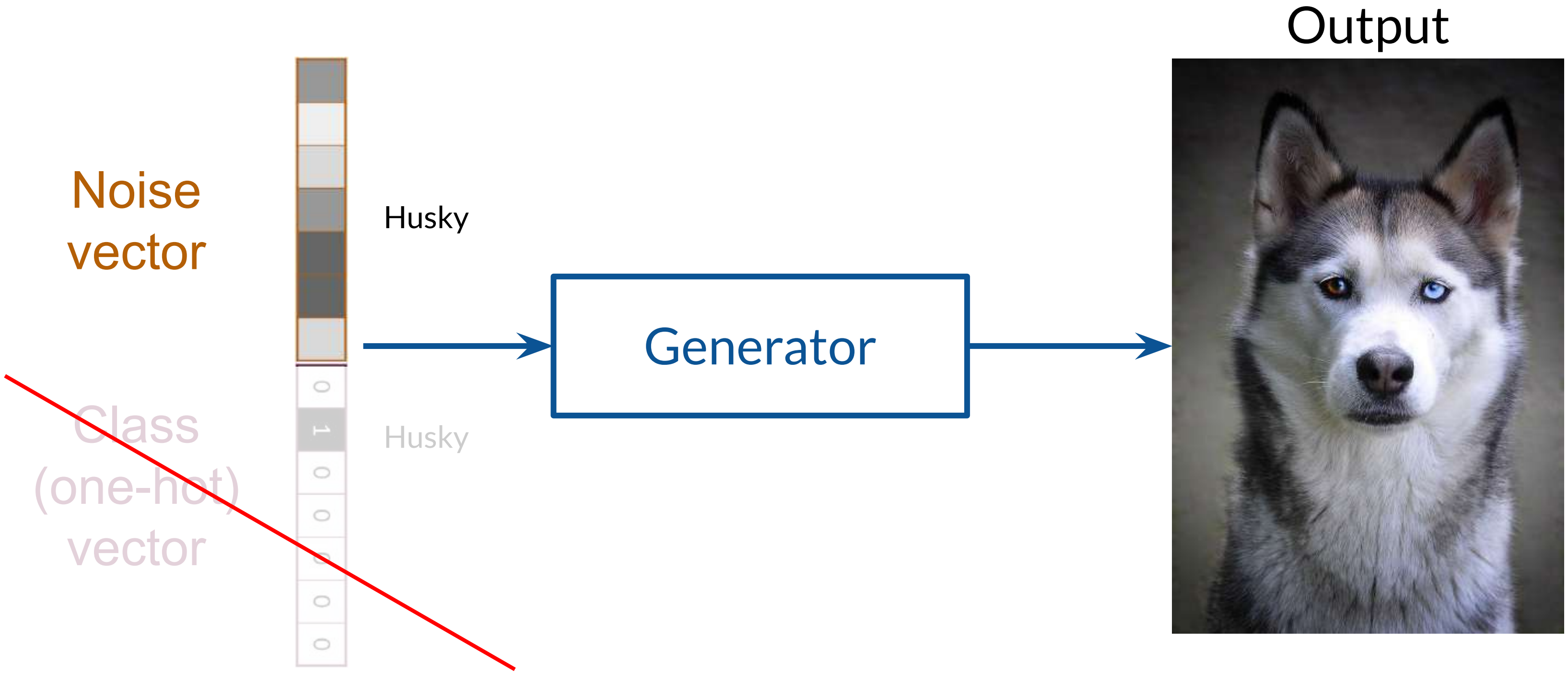
Changes to one feature don't affect the others



Encourage Disentanglement: Supervision



Encourage Disentanglement: Supervision



Encourage Disentanglement: Supervision

Difficult for continuous classes!

Noise vector

Husky

Generator

~~Class (one-hot) vector~~

~~0 1 0 0 0 0 0~~

~~Husky~~

Output

Encourage Disentanglement: Loss Function

$$v_1 = [\ 1,\ 2,\ 3, \ldots\ ]$$

$$v_2 = [\ 5,\ 6,\ 7, \ldots\ ]$$

Encourage Disentanglement: Loss Function

$$L_{\text{new}} = \underbrace{\boxed{L_{\text{original}}}}_{\text{Original loss}} + \underbrace{\boxed{\text{reg}_d}}_{\text{Regularization}}$$

$$v_1 = [\ 1, 2, 3, \ldots\ ]$$

$$v_2 = [\ 5, 6, 7, \ldots\ ]$$

Encourage Disentanglement: Loss Function

$$L_{\text{new}} = \underbrace{\boxed{L_{\text{original}}}}_{\text{Original loss}} + \underbrace{\boxed{\text{reg}_d}}_{\text{Regularization}}$$

$$v_1 = [\ 1,\ 2,\ 3, \ldots\ ]$$

$$v_2 = [\ 5,\ 6,\ 7, \ldots\ ]$$

Can be any loss function
(e.g. BCE, W-Loss)

Encourage Disentanglement: Loss Function

$$
\begin{array}{ccc}
 & z_1 & z_2 & & \\
v_1 = [& 1, & 2, & 3, & \ldots] \\
v_2 = [& 5, & 6, & 7, & \ldots]
\end{array}
$$

$$L_{\text{new}} = \underbrace{\boxed{L_{\text{original}}}}_{\text{Original loss}} + \underbrace{\boxed{\text{reg}_d}}_{\text{Regularization}}$$

Can be any loss function
(e.g. BCE, W-Loss)

Encourage Disentanglement: Loss Function

$$\begin{array}{ccc} & z_1 & z_2 & & \\ v_1 = [& 1, & 2, & 3, & \dots] \\ v_2 = [& 5, & 6, & 7, & \dots] \end{array}$$

Output feature #1

Output feature #2

$$L_{\text{new}} = \boxed{L_{\text{original}}} + \boxed{\text{reg}_d}$$

Original loss

Regularization

Can be any loss function
(e.g. BCE, W-Loss)

Summary

- Disentangled *Z*-spaces let you control individual features by corresponding *z* values directly to them
- There are supervised and unsupervised methods to achieve disentanglement