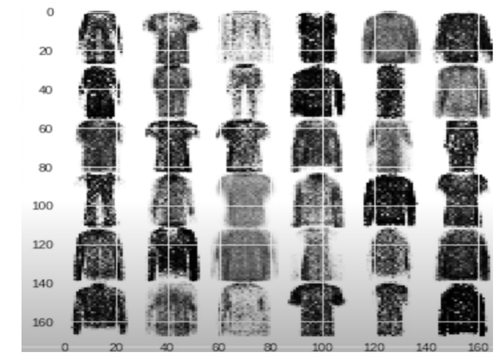
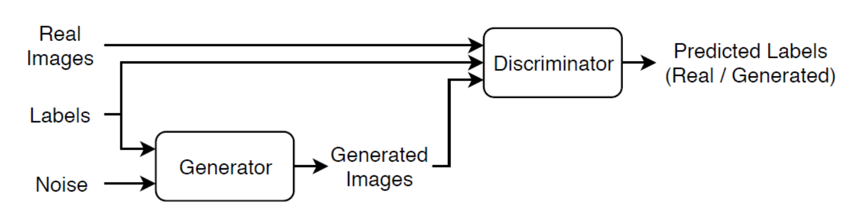
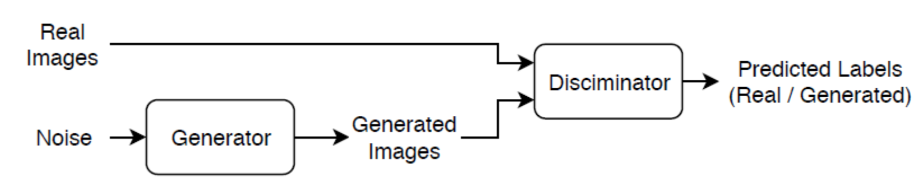
Day 14 Conditional GANs (cGANs)

- **Intro**
  - Look at an output of a GAN
    - Eg: Instead of producing images of clothes in general, it will produce images of pants, shirts, jackets, based on the provided label

      
    - No control over the type of output
  - Enables more precise generation and discrimination of images
  - How? Using "labels"
    - Class Labels - pieces of information that specify the data produced by the generator and the discriminator.
      - This will help them arrive at the desired results more quickly
    - They guide the Generator to generate more specific information
    - The labels help the Discriminator better distinguish between real and fake images
- **Applications**
  - Image-to-image translation
    - Allow images to evolve by considering labels
  - Creating images from text
    - Create high-quality photos based on text
    - Text and the richness of its vocabulary enables the generation of much more precise synthetic images
  - Video generation
    - Predict future frames of a video based on a selection of previous images
  - Face generation
    - Generate images of faces with specific attributes, such as hair or eye color.
  - Let's look at an example that we can understand better - MNIST handwritten digits
    - A simple DCGAN wouldn't let us choose the class of digits we are generating
  - Let's say we have an imbalanced dataset, and we want to generate more examples to balance it
    - Instead of collecting new data, we can train a cGAN to get data for the specific class requires
- **Architecture & Working**
  - **Conditioning**
    - Normal GAN can be made into a cGAN by providing additional info (y) to both generator and discriminator.
  - **GAN vs cGAN**
    - cGAN

    - GAN

  - **Generator**
    - Takes both the prior input noise (z) and the additional information (y) as inputs
    - Combined in a joint hidden representation, and the generator produces synthetic samples.
  - **Discriminator**
    - Takes both real data (x) and the additional information (y) as inputs
  - **Loss Function**
    - Exact same thing except, instead of single probability, it will be conditional probability
    - cGAN

      $$E\_x[\log (D(x|y))] + E\_z[\log (1 - D(G(z|y)))]$$
    - GAN

      $$E\_x[\log (D(x))] + E\_z[\log (1 - D(G(z)))]$$
    - E_x : Expected value wrt real data distribution
    - E_z : Expected value wrt noise data distribution
    - Objective of
      - Generator - minimize
      - Discriminator - maximize
- **Simply put...**
  - Conditional GANs are all about control
  - In normal GANs we aren't able to specify any conditions for G and D to look for
    - Therefore, users don't have any control over the network
  - By allowing specification of characteristics, cGANs solve this problem
- **Hands-on**