

簡単に体験できるGAN画像生成

ソニーネットワークコミュニケーションズ株式会社

著作権情報

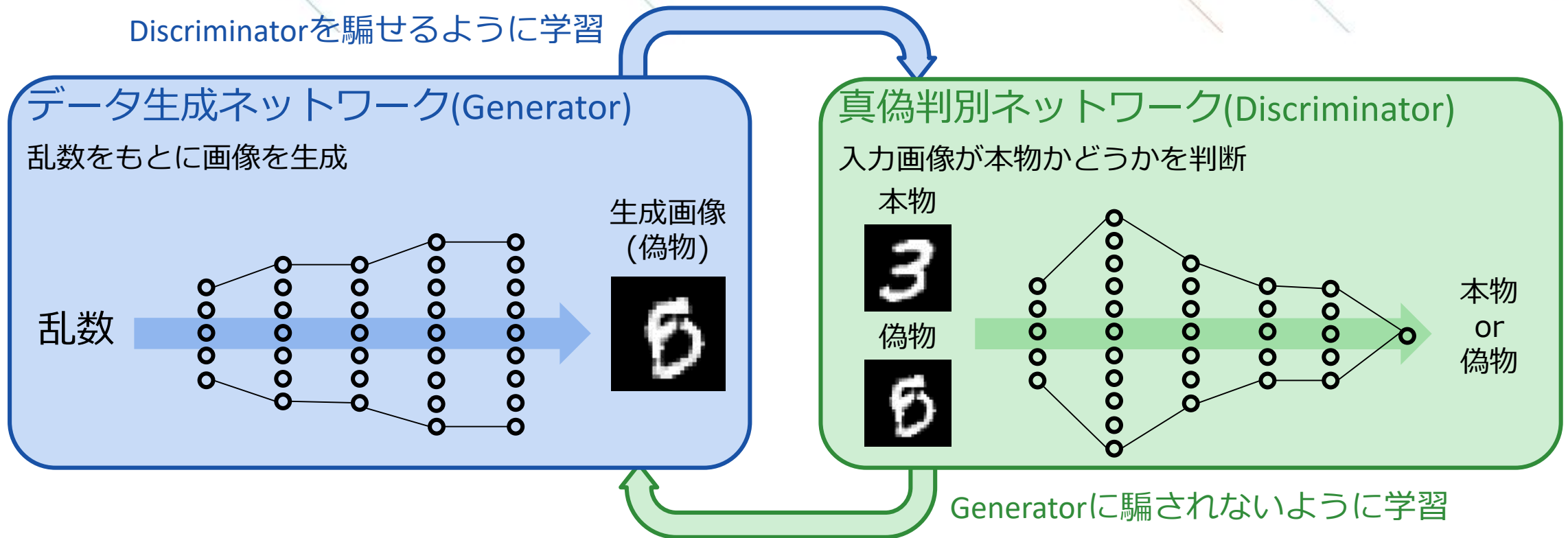
本資料は著作権法で保護された著作物であり、その全部または一部を許可なく複製すること、複製物を配布すること、他のコンピューター用へ変換することおよび他の言語へ翻訳することは著作権の侵害です。

商標表示

本資料に記載されている会社名、製品名およびサービス名は各社の商標または登録商標です。

GAN(Generative Adversarial Networks)とは

「本物らしさ」を学習することにより、高品質なデータ生成などを実現するテクニックです。
データを作るネットワークと本物かどうかを見極めるネットワークの2つを敵対させて同時に学習することで、より本物らしいデータを作るネットワークを作成することができます。



Generative Adversarial Networks, Ian J. Goodfellow et al., <https://arxiv.org/abs/1406.2661>

サンプルプロジェクトの起動

以下のリンクをクリックし、Neural Network ConsoleにあるGANのサンプルプロジェクトを起動します。

GANのサンプルプロジェクト: https://dl.sony.com/console/#/project?project_id=9

※アカウント未取得の方や、アカウント取得済みの方でもセッションが切れている方は、以下の画面に遷移します。この場合、アカウント取得やサインイン手続きを実施後に、もう一度上のリンクをクリックください。



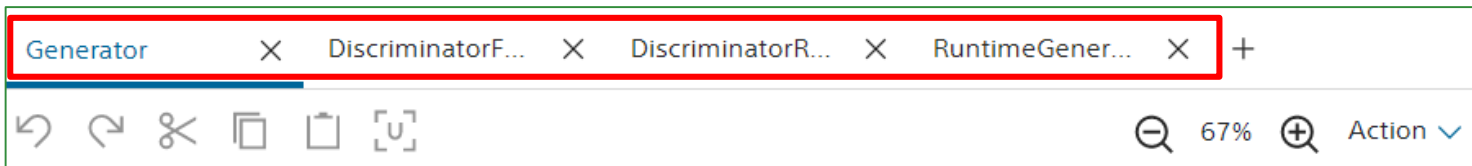
サインイン手続きの詳細はこちら: <https://support.dl.sony.com/docs-ja/アカウントの作成およびサインイン/>

全体概要

ネットワークを学習するためのネットワークページ3つと学習後に画像を生成するためのネットワークページ1つから構成されます。

ネットワークの準備

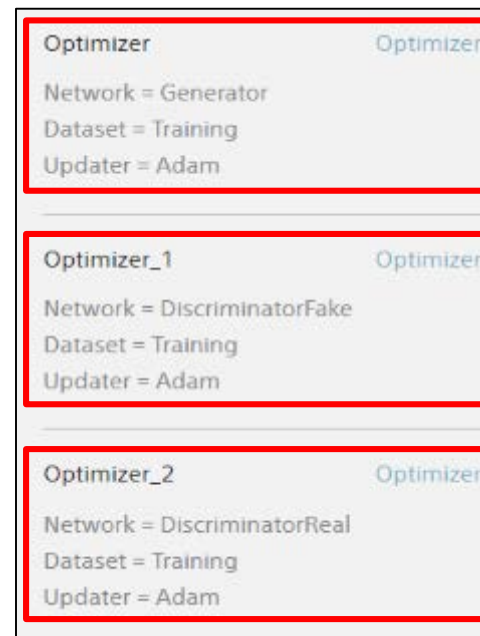
EDITタブに4つのネットワークタブが存在
ネットワークタブ間のレイヤー名は同一であり、パラメータは共有される



- Generator
画像を生成するネットワークを学習
- DiscriminatorFake
生成画像が偽物であることを学習
- DiscriminatorReal
入力画像が本物であることを学習
- RuntimeGenerator
推論用の画像を生成するネットワーク

Optimizerの設定

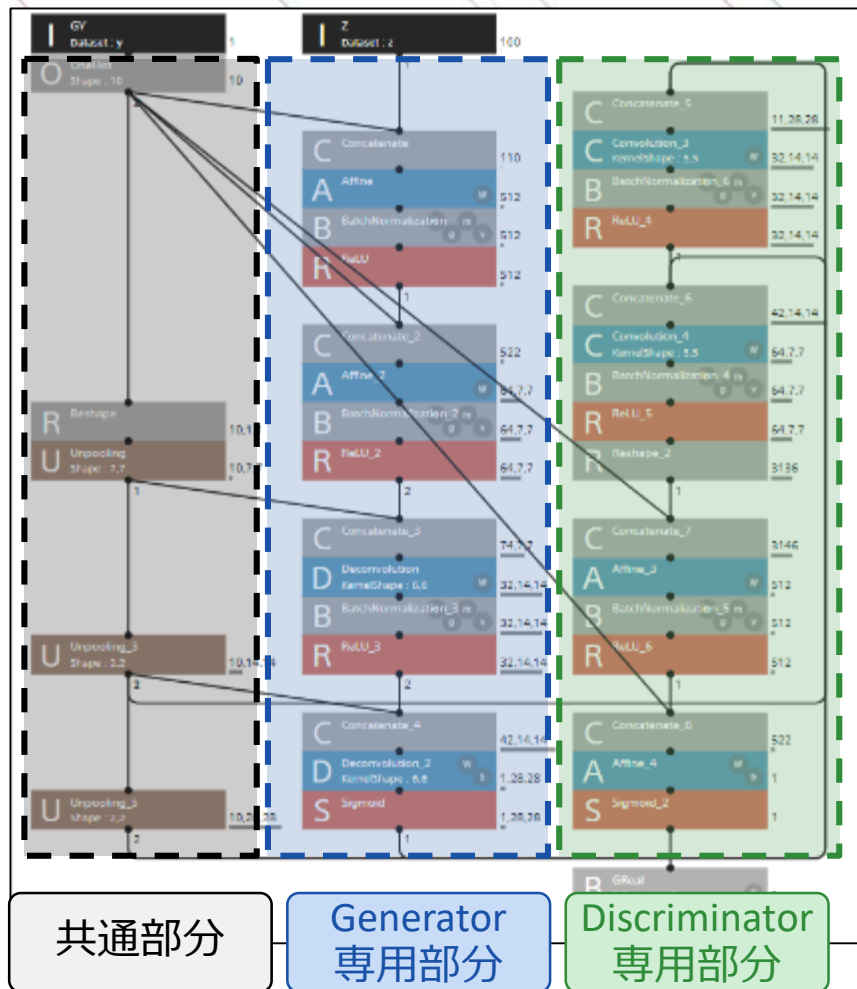
Configタブに学習用のネットワークの全てに対して学習のためのアルゴリズムをセット



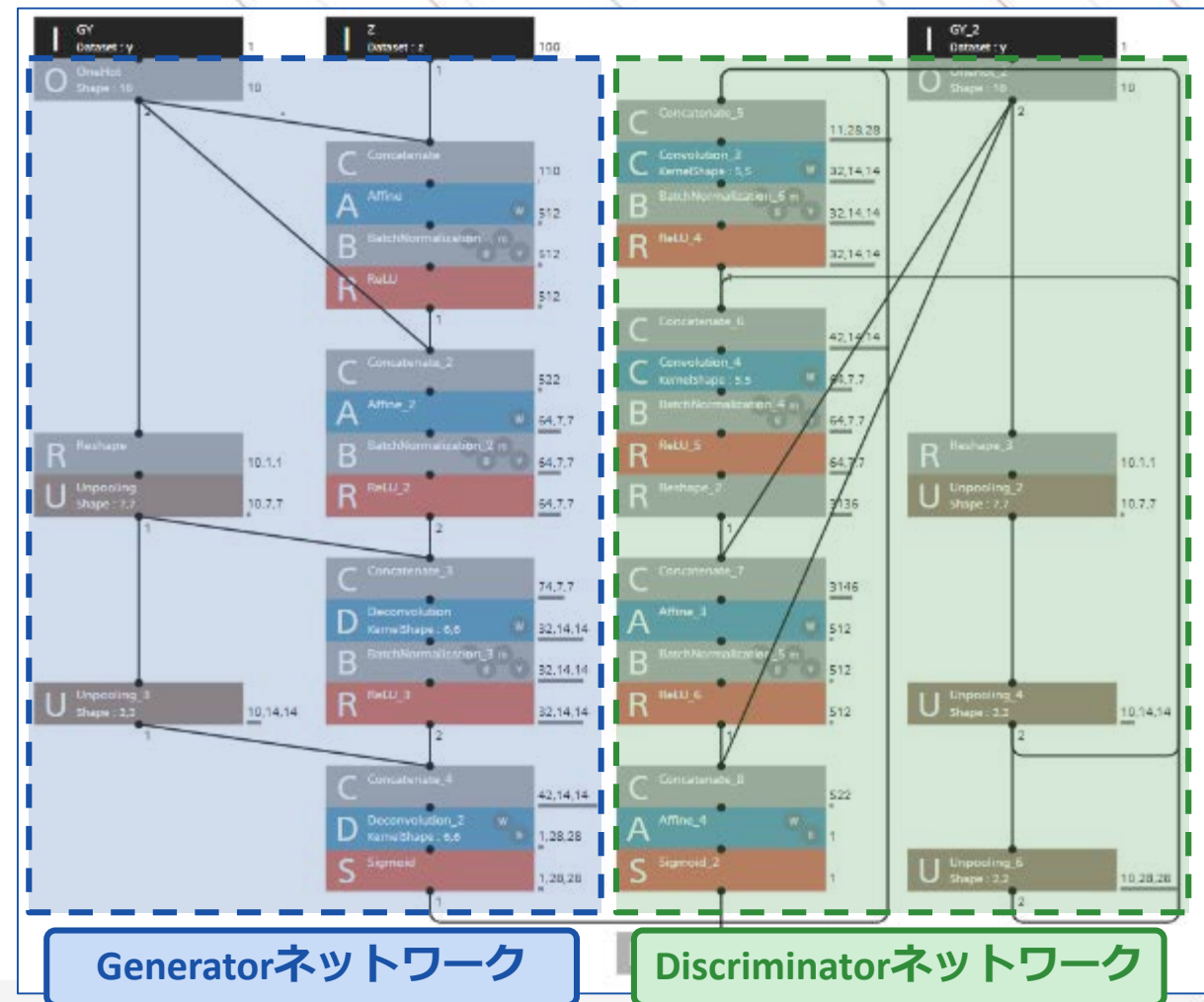
ネットワークタブの共通部分

左側の列はラベル情報からGeneratorとDiscriminatorの専用部分の入力情報を生成しています。

したがって、GeneratorとDiscriminatorのネットワークは左側列を含んだものになります。



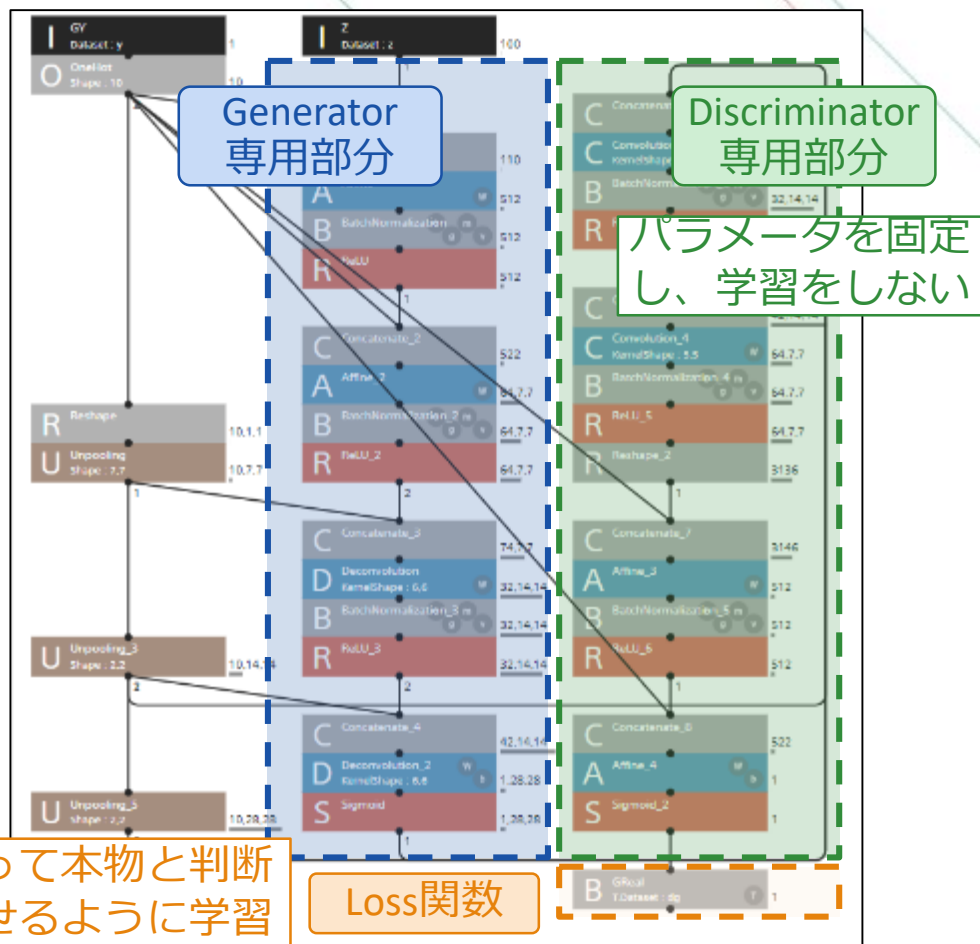
共通部分を
展開すると



Generatorタブ

Generatorタブでは画像を生成するGeneratorネットワーク部分の学習をします。

Discriminator部分はパラメータを固定し学習せず、誤って本物と判断するように学習させます。



パラメータの固定

- ✓ Discriminator部分のレイヤーの中のパラメータの学習率を0にすることで、学習させず、パラメータを固定

ParameterScope	Convolution_3
W.File	
W.Initializer	NormalConvolut
W.InitializerMultiplier	1
W.LRateMultiplier	0

学習率を0に設定

Loss関数の設定

- ✓ BinaryCrossEntropyで1(本物)にどの程度近いかで誤差を定義

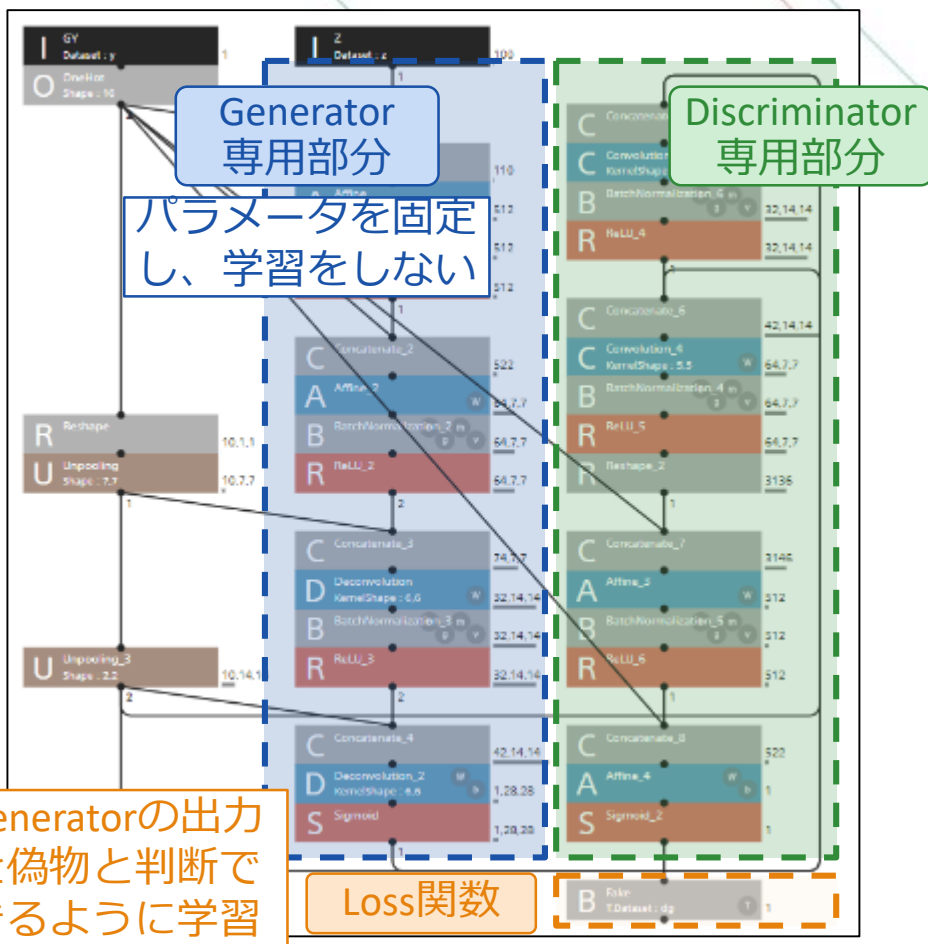
Name	GReal
Input	1
T.Dataset	dg
T.Generator	Constant
T.GeneratorMultiplier	1

Constantで定数1にしてターゲットに設定

DiscriminatorFakeタブ

DiscriminatorFakeタブではDiscriminatorネットワーク部分の学習をします。

Generator部分はパラメータを固定し学習せず、Generatorの出力を偽物と判断するように学習させます。



パラメータの固定

- ✓ Generator部分のレイヤーの中のパラメータの学習率を0にすることで、学習させず、パラメータを固定

A Affine

ParameterScope	Affine
W.File	
W.Initializer	NormalAffineGl
W.InitializerMultiplier	1
W.LRateMultiplier	0

学習率を0に設定

Loss関数の設定

- ✓ BinaryCrossEntropyで0(偽物)にどの程度近いかで誤差を定義

B BinaryCrossEntropy

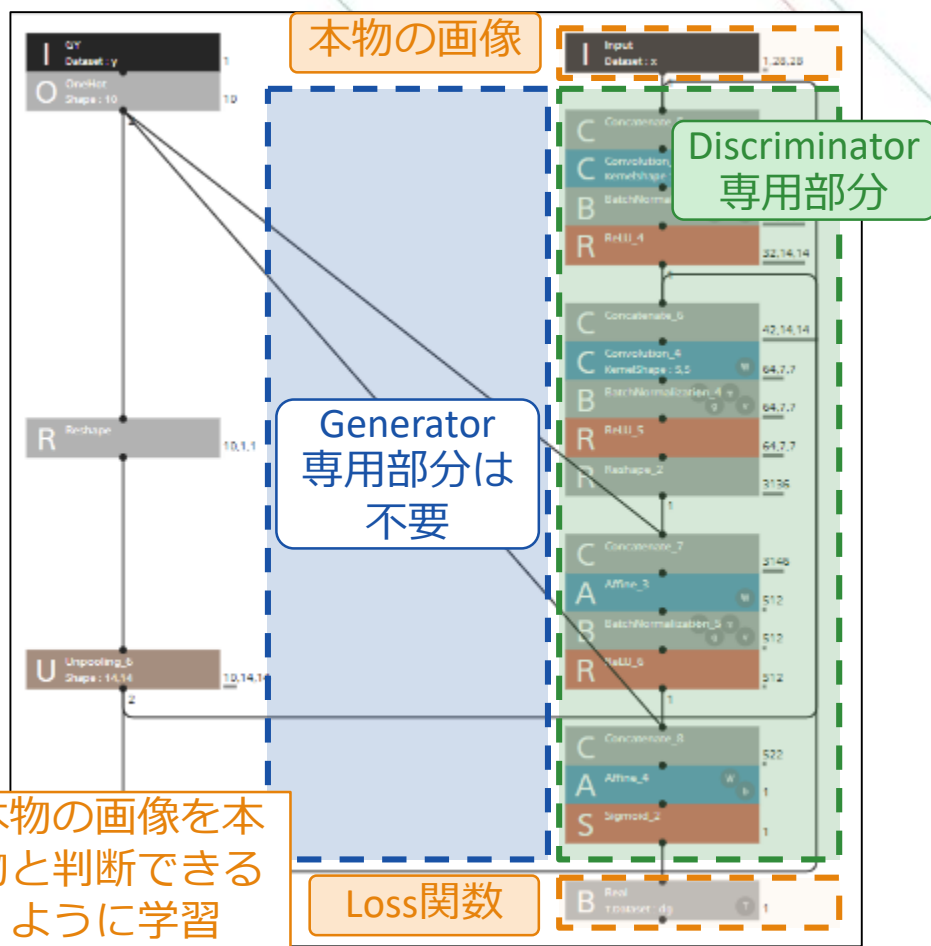
Name	Fake
Input	1
T.Dataset	dg
T.Generator	Constant
T.GeneratorMultiplier	0

Constantで定数0にし、ターゲットを0に設定

DiscriminatorRealタブ

DiscriminatorRealタブではDiscriminatorネットワーク部分の学習をします。

Generator部分は利用せず、本物の画像を入力にして本物と判断するように学習させます。



Loss関数の設定

- ✓ BinaryCrossEntropyで1(本物)にどの程度近いかで誤差を定義

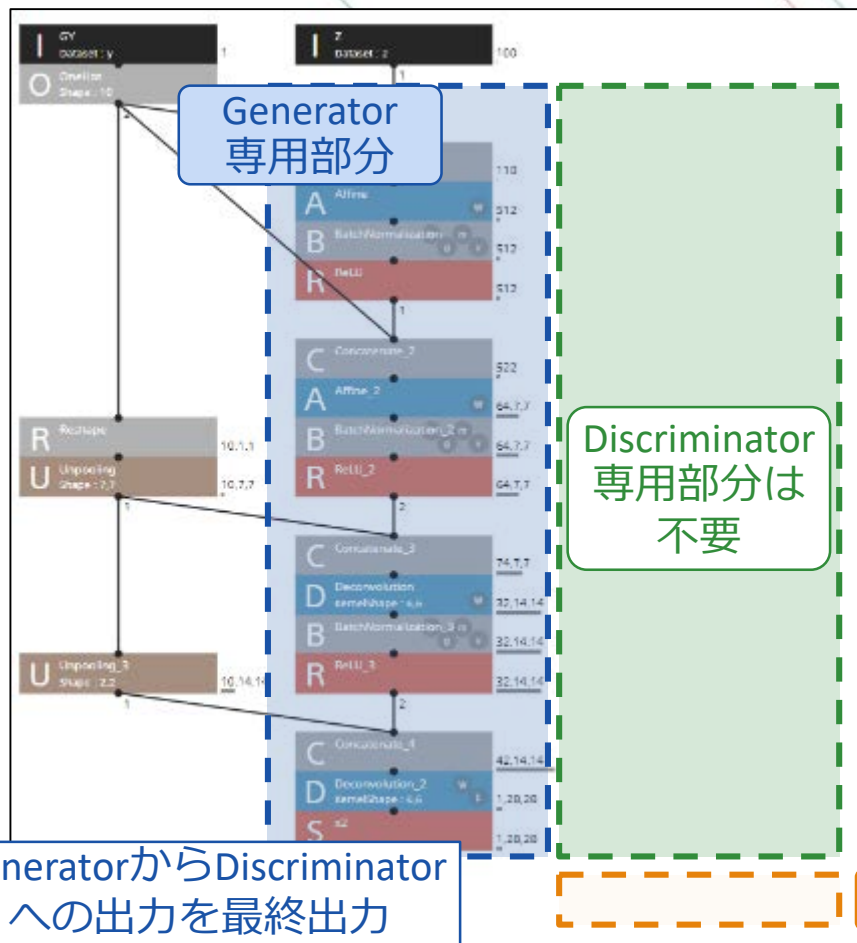
B BinaryCrossEntropy

Name	Real
Input	1
T.Dataset	dg
T.Generator	Constant
T.GeneratorMultiplier	1

Constantで定数1にし、ターゲットを1に設定

RuntimeGeneratorタブ

推論用にGenerator部分だけを抜き出したネットワークです。
乱数と生成する画像のラベルをもとに生成画像を出力します。



GAN設計、学習時の注意点

- Discriminatorのパラメータは共有されるようにする

Neural Network Consoleの場合、ParameterScopeプロパティに共通の文字列を指定することでパラメータを共有できる

- Generator学習時はDiscriminatorのパラメータを、Discriminator学習時はGeneratorのパラメータをそれぞれ固定する

Neural Network Consoleの場合、Convolution、Affine、DeconvolutionのW.LRateMultiplier、b.LRateMultiplier、BatchNormalizationのbeta.LRateMultiplier、gamma.LRateMultiplierプロパティを0に設定することでパラメータを固定できる

- ロスが最も小さくなった点が最も生成品質が高いとは限らない

Generatorのロスが小さくなると、Discriminatorのロスが大きくなる敵対関係にあるため

Neural Network Consoleの場合、CONFIGタブのSave Bestチェックボックスを外すことで、最終世代のモデルが保存されるようにする

- 安定した学習のためには様々な工夫が必要

とりあえずの設計でもそれなりに安定して学習できるCNNなどと異なり、GANはGenerator、Discriminatorのどちらかに精度が偏学習が進まなくなる、特定の画像しか生成されなくなる（Mode Collapse）などの現象に陥ることも多い

現在も安定性や品質を向上させる様々なテクニックが提案され続けている



SONY

SONYはソニー株式会社の登録商標または商標です。

各ソニー製品の商品名・サービス名はソニー株式会社またはグループ各社の登録商標または商標です。その他の製品および会社名は、各社の商号、登録商標または商標です。