MiniBatchを実装できないので、Batchの実装時点で正確な動作が確認されたpt2において、もう一度誤差計算の実装方法を確認した。その結果、改めてNNにおける前方向の誤差計算は、

Loss = self.Function.forward(data, comp\_data) / minibatch

data：NNの前方向出力、comp\_data：正解データ、minibatch：ミニバッチサイズ

という方法で行われていることを確認した。ここで、pt2におけるminibatchは1であり、iteration数はbatch sizeと同数である。さらに、Lossはその後batch数と同数だけself.LossSumという変数に保存されることが分かった。また、同時に誤差の計算に使用された、NNの前方向出力も同様にself.ResultSumという変数に保存していることが分かった。

self.ResultSum：出力データのバッファ

self.LossSum ：誤差÷minibatchのバッファ

1エポック終了後の誤差は、次のように計算される。

1. self.ave\_loss関数を用いてBatchResultおよびBatchLossの計算を行う。

BatchResult = sum(self.ResultSum) / batch\_size

出力の合計 ÷ バッチサイズ

BatchLoss = sum(self.LossSum) / batch\_size

(誤差 / minibatch)の合計 ÷ バッチサイズ

1. その後、フィードバック信号を出力するためにself.Function.backward関数による行う。

BackSignal = self.Function.backward(BatchResult, BatchLoss)

BackSignal：フィードバック信号

1. Self.Function.backward関数内部では、次のような計算が行われている。

def backward(self, y, t, dout = 1):

if dout == 1:

if y.size == t.size:

batch\_size = t.shape[0]

if batch\_size == 1:

dout = y - t

else:

dout = (y - t) / batch\_size

return dout

　この手順をたどると、誤差の計算はminibatchの際に1回、フィードバック信号の計算で1回、計2回行っていることになる。具体的には、minibatchの際は1回の出力ごとの誤差を、フィードバック信号計算の際は、batch分を平均したNN出力とbatch分を平均した誤差との誤差を計算している。

実験1

　本などによると、フィードバック信号は「誤差の合計÷batch数」となっていたため、self.Function.backward関数の一部を以下のように変更し、結果がどう変わるか実験を行った。

def backward(self, y, t, dout = 1):

dout = t

return dout

ここでt = BatchLossである

実験の条件

* epochs = 120
* BatchSize = 128
* レイヤ

1. 入力層
2. 隠れ層1（全結合50ユニット ReLU関数）
3. 隠れ層2（全結合50ユニット ReLU関数）
4. 出力層 （全結合1ユニット 恒等関数 ）

* 誤差の計算：2乗和誤差

結果1

　プログラム変更前の結果と、変更後の結果を以下に示す。

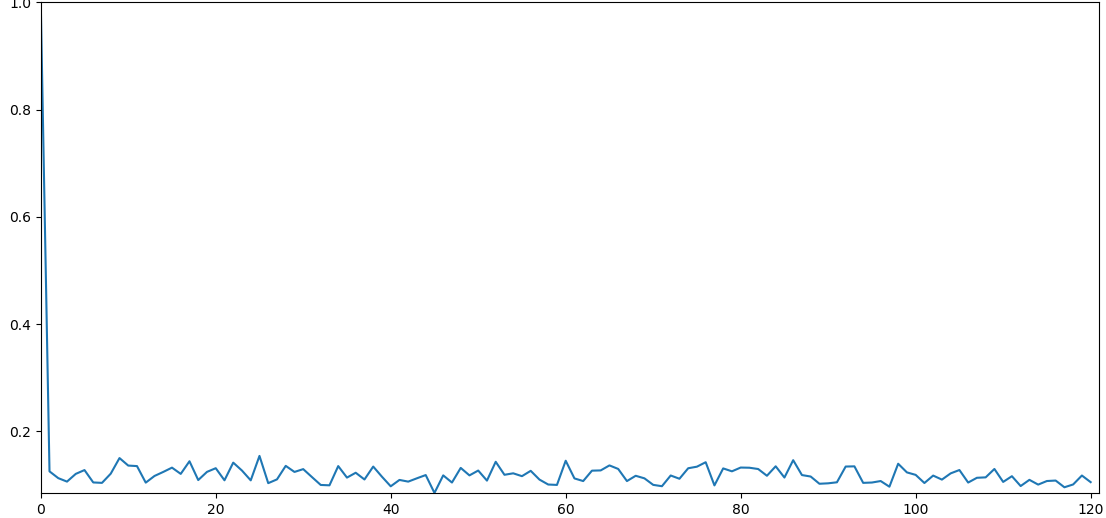


図 　プログラム変更前

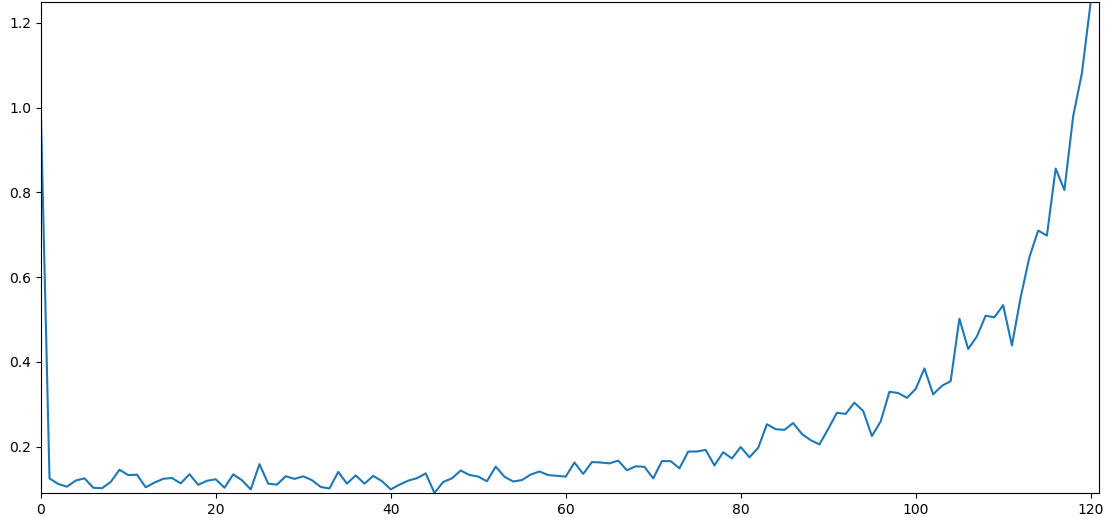


図 　プログラム変更後

図1では、問題なく誤差が下がった。しかし図2では、50回を迎えたあたりから誤差が上昇を始めた。

プログラムの変更

ここで、上述のフィードバック信号を計算する手順において、出力の誤差による修正に明らかな誤りがあったことがわかった。というのも、「batch分を平均したNN出力とbatch分を平均した誤差との誤差を計算している」ことから、フィードバックによる収束が誤差のbatch平均に基づいて行われているため、正解データに向けて収束していない可能性が出てきた。

そこで、昨日作成した誤差計算用のクラスであるLossクラスを無効化し、メインプログラムを直接変更する形で修正を行った。具体的には、epoch毎に重みの平均値を計算し、その値を用いて重みを更新することにした。重みの平均化には、これまで分母にbatch sizeを用いてきたが、iterationを導入したことにより分母をデータの行数であるIRS(Input Row Size)とした。