論 文
41~3~8

수정된 Activation Function Derivative를 이용한 오류 역전파 알고리즘의 개선

Improved Error Backpropagation Algorithm using Modified Activation Function Derivative

權 熙 容*·黃 熙 隆**
(Hee-Yong Kwon · Hee-Yeung Hwang)

Abstract - In this paper, an Improved Error Back Propagation Algorithm is introduced, which avoids Network Paralysis, one of the problems of the Error Backpropagation learning rule. For this purpose, we analyzed the reason for Network Paralysis and modified the Activation Function Derivative of the standard Error Backpropagation Algorithm which is regarded as the cause of the phenomenon. The characteristics of the modified Activation Function Derivative is analyzed. The performance of the modified Error Backpropagation Algorithm is shown to be better than that of the standard Error Back Propagation algorithm by various experiments.

Key Words : Neural Network(신경망), Error Backpropagation(오류 역전파), Multilayer Perceptron (다층 퍼셉트론), Learning Rule(학습규칙), Network Paralysis(학습마비)

1. 서 론

다층 퍼셉트론 네트웍의 학습규칙으로 사용되고 있는 오류 역전파 (Error Backpropagation, EBP)는 모델의 단순함과 강력한 문제해결 능력으로 많은 응용분야에서 이용되고 있다. 그러나 이 방법은 경사 추적법을 기본으로 하고 있어 학습속도가 매우 느리고 local minima에 빠질 위험성이 있어 문제로 되고 있다. 최근 이를 해결하기 위해 여러 가지 연구가 행해지고 있으며 많은 연구결과가 나오고 있다. [1, 2, 4, 5, 8] 특히 학습도중 네트웍이 큰 오류 값을 가짐에도 불구하고 학습이 진전되지 않는 Network Paralysis 현상에 대해 그 원인이

*正 會 員 : 서울大 大學院 컴퓨터工學科 博士課程
**正 會 員 : 서울大 工大 컴퓨터工學科 敎授·工博
接受日字 : 1990年 7月 3日
1次修正 : 1991年 12月 10日
2次修正 : 1991年 12月 24日

규명되기 시작함에 따라 이 현상을 방지하고 개선하기 위한 연구가 행해지고 있다. 그러나 이들은 현재 출력 층에서 만 수정을 하는 수준이다. 본 논문은 이와같은 Network Paralysis 현상을 기존의 EBP 학습규칙을 간단히 수정하여 은닉층을 포함하여 개선할 수 있는 새로운 알고리즘을 제안하고 있다.

우선 기존의 표준 EBP 모델에 대해 간략히 소개하고 이어서 이 표준 EBP 모델의 Error Surface를 검토하여 Network Paralysis 현상의 원인을 규명한다. 이 분석을 토대로 새로운 학습규칙을 제안하고, 제안된 규칙의 동작특성을 밝혀 Network Paralysis 현상을 방지할 수 있음을 보인다. 끝으로 다양한 실험을 통해 새로운 학습규칙의 성능이 우수함을 입증하였다.

2. 오류 역전파(EBP) 학습규칙

EBP 알고리즘은 다층 전방향 신경망(Multilayer Perceptron Network)에서 적용되는 학습 규칙으로서 multi dimensional mapping을 구현한다. [7] 네트워크의 구조는 processing element(unit)들이 층(layer)을 구성하고 각층간에는 fully connected network을 형성하고, 층내에서는 연결이 없는 것으로 가정한다.

이때 EBP 알고리즘은 네트워크의 Performance를 Error Measure Function(Cost Function)으로 표현하고 이 함수를 최소화 하는 연결 가중치를 찾는다. 이때 Error Measure Function은 최소 자승법 (Least Mean Square, LMS)으로, 최소화는 경사 추적법을 이용하여 구현된다. 즉 EBP학습규칙은

unit j에 대한 network total input

$$\text{net}_{pj}=\sum_i w_{ji}o_{pi}$$

semilinear activation function

$$o_{pj}=f_j(\text{net}_{pj})=\frac{1}{1+e^{-\text{net}_{pj}}}$$

error measure function

$$E=\sum_p E_p=\frac{1}{2}\sum_p\sum_j(t_{pj}-o_{pj})^2$$

을 가정하고 weight를 변화시켜 E를 최소화하는 것이다.

그러므로 E를 최소화하는 방향 $-\partial E/\partial w_{ji}$, 즉 Gradient Descent를 구해 그반대 방향으로 연결 가중치를 변화시킴으로서 학습을 진행한다. [3]

이를 위해 $-\partial E/\partial w_{ji}$를 구해보면 EBP알고리즘은 다음과 같이 요약될 수 있다.

$$\Delta_p w_{ji}=\eta(-\partial E/\partial w_{ji})=\eta\delta_{pj}o_{pi} \quad (1)$$

여기서

$$\delta_{pj}=(t_{pj}-o_{pj})f'_j(\text{net}_{pj}) \text{ for output units} \quad (2)$$

$$\delta_{pj}=\sum_k\delta_{pk}w_{kj}f'_j(\text{net}_{pj}) \text{ for hidden units} \quad (3)$$

$$f'(\text{net}_{pj})=o_{pj}(1-o_{pj}) \quad (4)$$

3. Network Paralysis 현상

위에서 살펴본 EBP 학습규칙은 Network Paralysis와 local minima로 인해 학습에 실패할 수 있고 또한 수렴은 무한히 작은 연결 가중치 조정의 반복에 의해서만이 보장되고 있다. 그러므로 학습 속도가 매우 느린 문제가 있다.

특히 본 논문에서 해결하고자 하는 Network Paralysis현상은 네트워크의 출력이 원하는 출력과 거리가 있는 상태, 즉 오류가 큰 상태임에도 불구하고 학습이 거의 중단되는 상태로서 $\Delta w\approx 0$인 상태라고 할 수 있다. [6]

이 같은 현상의 원인을 밝히기 위해 우선 다음과 같이 EBP 학습규칙을 재구성한다.

$$\Delta_p w_{ji}=\eta\delta_{pj}o_{pi}$$

$$\delta_{pj}=T_E T_S$$

여기서

$$T_E=(t_{pj}-o_{pj}) \quad \text{for output units}$$

$$=\sum_k\delta_{pk}w_{kj} \quad \text{for hidden units}$$

$$T_S=f'_j(\text{net}_{pj})=o_{pj}(1-o_{pj})$$

여기서 T_S는 unit의 stability를 표현하는 항으로서 역전파 되어야 할 오류의 양을 나타내며 그 unit의 stability를 나타낸다. T_E는 오류의 방향 즉 gradient의 방향을 나타 낸다.

위 식에서 Δw를 0인 상태, 즉 학습을 멈추게 하는 현상은 다음과 같은 경우에 발생한다.

i) $T_S\approx 0$: unit의 출력이 0 또는 1에 가까운 경우

ii) $T_E\approx 0$: 오류가 0에 가까운 경우

iii) $o_{pi}\approx 0$: weight에 연결된 입력 o_{pi}가 0에 가까운 경우

이때 ii)와 iii)으로 인한 $\Delta w\approx 0$인 상태는 바람직한 상태이므로 문제가 되지 않는다. i)의 경우 $T_E\approx 0$이면서 $T_S\approx 0$이면 unit의 출력이 0이나 1에

가까운 경우로 target과 출력이 일치하는 출력을 내고 있는 unit라고 할 수 있다. 따라서 이 unit에 연결된 연결 가중치는 학습이 중지되어 안정된 상태를 유지한다. 이는 바람직한 결과가 된다.

그러나 $T_E \gg 0$이면서 $T_S \approx 0$이면 오류가 커도 $\delta \approx 0$이 되고 따라서 $\Delta w \approx 0$이 되어 학습이 중단된다. 이같은 현상은 이 unit의 출력이 Activation Function의 양 극단에 치우쳐 있는 경우로 하위층과의 연결 가중치가 매우 크거나 작은 경우에 발생한다. 이와같은 경우 Activation Function은 약간의 가중치 변화로는 출력에 영향을 주지 못하게 되는 포화상태로 된다. 따라서 잘못된 방향으로 포화된 unit는 자신에 연결된 가중치를 변화시킬 수 없고, 또한 하위층에서 네트워크 입력을 변화시켜도 포화상태에서 오랫동안 빠져나오질 못한다. 이때 네트워크은 더이상의 학습이 불가능한 Network Paralysis현상에 빠지게 된다. [5, 8]

그러므로 Network Palaysis현상은 $T_E \gg 0$이면서 $T_S \approx 0$인 경우라고 할 수 있다.

4. 개선된 EBP

위의 결과를 볼때 Network Paralysis 현상과 학습 속도 둔화는 T_S 즉 stability항이 target를 0으로 만드는데 그 원인이 있다. 따라서 이와같은 현상을 방지하기 위해서는 unit의 stability를 결정할 때 시스템 오류를 반영할 수 있는 인자를 포함시켜 원치않는 포화 상태인 경우 target쪽으로 학습이 되도록 하는 방법이 필요하다. 즉 오류가 크면 T_S를 크게하고 작으면 작게할 필요가 있다. 이와같은 조건을 만족시키기 위해서는 그림 1과 같은 표준 Activation Function Derivative의 모양을 그림 2와 같은 모양으로 수정하여야 한다. 이 결과는 여러가지 수식으로 표현이 가능하지만[5] 원래의 Activation Function Derivative의 포물선 모양을 유지하면서 식의 단순화를위해 Activation Value를 target과의 평균으로 수정하여 다음과 같이 개선된 T_S를 제안한다.

$$T'_S = \left[\frac{t_{pj} + o_{pj}}{2}\right]\left[1 - \frac{t_{pj} + o_{pj}}{2}\right] \tag{5}$$

T'_S의 동작 특성을 다음과 같이 두가지 경우로 나누어 분석한다.

4.1 output unit인 경우〈그림 2〉

if $t_{pj} \approx 0$,

$$T'_S = \left[\frac{o_{pj}}{2}\right]\left[1 - \frac{o_{pj}}{2}\right] \tag{6}$$

if $t_{pj} \approx 1$,

$$T'_S = \left[\frac{1 + o_{pj}}{2}\right]\left[\frac{1 - o_{pj}}{2}\right] \tag{7}$$

이 식을 그림으로 보인 것이 그림 2이다. 즉 target과 실제 출력이 가까우면 Activation Function Derivative는 0에 근접하여 델타를 0에 가깝게 만들고 따라서 학습이 적게 행해진다. 그러나 target과 출력이 멀어지면 Activation FunctionDerivative는 최대화하여 델타를 크게 만들고 따라서 학습이 많이 이루어 지도록 한다.

4.2 hidden unit인 경우〈그림 3〉

hidden unit의 target을 알 수 없으므로 제안된 식은 hidden unit에 대해 적용이 곤란하다. 그러나

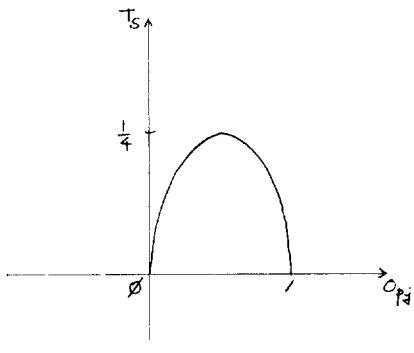

그림 1 표준 Activation Function Derivative
Fig. 1 Standard Activation Function Derivative

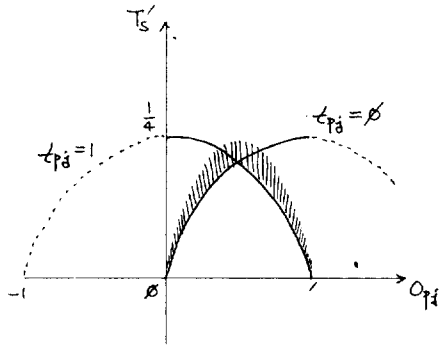

그림 2 출력층의 수정된 Activation Function Derivative
Fig. 2 Modified Activation Function Derivative in Output Layer

표 1 식(8)의 동작특성
Table 1 Characteristics of eq. 8

a	max(T'_s)	min(T'_s)
$a<-0.5$	$-a(1+a)$	$a(1-a)$
$-0.5<a<0$	0.25	$a(1-a)$
$0<a<0.5$	0.25	$-a(1+a)$
$a>0.5$	$a(1-a)$	$-a(1+a)$

$a=1/2\sum\delta_{pk}w_{kj}$
$0<o_{pj}<1$

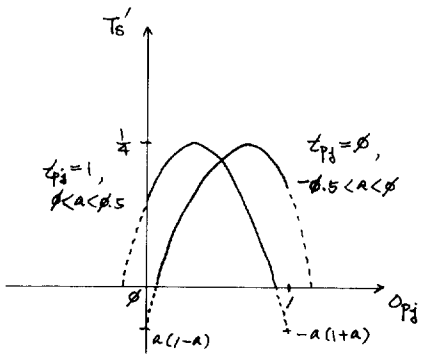

그림 3 은닉층의 수정된 Activation Function Derivative
Fig. 3 Modified Activation Function Derivative in Hidden Layer

다음과 같이 추정된 target을 가정하면 제안된 식을 hidden unit에도 적용할 수 있다. 이 경우 추정된 target이 계속 변화하기는 하지만 학습이 진행됨에 따라 $\sum\delta_{pk}w_{kj}$가 0에 가깝게 되므로 추정된 target는 unit의 출력과 같은 방향으로 변화한다. 이렇게 되면 T'_s는 T_S와 같은 특성을 가지므로 가정은 타당성을 가질 수 있다. 유도과정은 다음과 같다.

$$T_E=\sum\delta_{pk}w_{kj}=\hat{t}_{pj}-o_{pj}$$
($\hat{t}$는 target에 대한 가정치)
$$\hat{t}=o_{pj}+\sum\delta_{pk}w_{kj}$$
$$\therefore T'_s=\left[\frac{t_{pj}+o_{pj}}{2}\right]\left[1-\frac{\hat{t}_{pj}+o_{pj}}{2}\right]$$
$$=\frac{2o_{pj}+\sum\delta_{pk}w_{kj}}{2}\left[1-\frac{2o_{pj}+\sum\delta_{pk}w_{kj}}{2}\right]$$
$$=\left[o_{pj}+\frac{1}{2}\sum\delta_{pk}w_{kj}\right]\left[1-\left[o_{pj}+\frac{1}{2}\sum\delta_{pk}w_{kj}\right]\right] \quad (8)$$

이 식의 동작특성을 정리하면 표 1과 그림 3과 같다. 이 결과는 그림 2에서 보이는 출력 unit의 경우와는 약간 다르지만, 오류가 크면서 출력이 0이나 1에 가까운 경우 큰 값을 생성시킨다는 점에서 같은 의미를 갖는다는 것을 알 수 있다.

그러나 그림 3에서 알 수 있듯이 식(8)은 $\sum\delta_{pk}w_{kj}$에 따라 $T'_s<0$으로 만들고 이는 Gradient의 방향에 영향을 주므로 바람직 하지 못하다. 따라서 윗 식을 항상 $T'_s>0$으로 하기위해 표 1과 그림 3을 참고로 하여 다음과 같이 재 수정한다.

$$T''_s=\left[o_{pj}+\frac{1}{2}\sum\delta_{pk}w_{kj}\right]\left[1-\left[o_{pj}+\frac{1}{2}\sum\delta_{pk}w_{kj}\right]\right]+b$$

이때 $T'_s>0$을 만족시키기 위한 b의 범위는 다음과 같다.

$$b=\begin{cases}-a(1-a) & a<0\text{인 경우}\\ a(1+a) & a>0\text{인 경우}\end{cases}$$
$$a=1/2\sum\delta_{pk}w_{kj}$$

이때 T'_s의 최대값은 다음과 같다.

$a<-0.5$이면
$$\max(T'_s)=(-a(1+a)+b)-(a(1-a)+b)=-2a(>1)$$
$a>0.5$이면
$$\max(T'_s)=(a(1-a)+b)-(-a(1+a)+b)=2a(>1)$$
$-0.5<a<0$이면
$$\max(T'_s)=(0.25+b)-(a(1-a)+b)=0.25-a+a^2$$
$$0.25<\max(T'_s)<1$$
$0<a<0.5$이면
$$\max(T'_s)=(0.25+b)-(-a(1+a)+b)=0.25+a+a^2$$
$$0.25<\max(T'_s)<1$$

그러므로 T'_s는 항상 양의 값을 갖고 경사 방향에 영향을 주지 않는다. 단 $|a|>0.5$이면 T'_s가 1보다 커져 증폭작용을 하므로 $|a|<0.5$로 한정시켜 T_S를 증폭시키지 않도록 한다. 실제로 EBP 학습규칙의 유도과정은 매우 작은 양의 오류 역전파를 전제로 하고 있으므로 이와같은 제한은 일반성을 잃지 않으며, 실험중에도 발생하지 않았다.

이상과 같은 T'_s의 분석을 다시 정리하면, 수정된 학습규칙은 다음과 같다.

$$\Delta_p w_{ji} = \eta \delta_{pj} o_{pi}$$

$$\delta_{pj} = (t_{pj} - o_{pj})\left[\frac{t_{pj}+o_{pj}}{2}\right]\left[1-\frac{t_{pj}+o_{pj}}{2}\right] \quad \text{for output units}$$

$$= \sum_k \delta_{pk} w_{kj}((o_{pj}+a)(1-(o_{pj}+a))+b) \quad \text{for hidden units}$$

여기서, $-0.5 < a < 0.5$

$$a = 1/2 \sum \delta_{pk} w_{kj}$$

$$b = \begin{cases} -a(1-a) & a<0 \\ a(1+a) & a>0 \end{cases}$$

이상과 같이 새로이 정의된 학습규칙은 기존의 표준 EBP가 갖는 경사 추적 학습특성은 그대로 유지하면서 역전파되는 오류의 양을 target에 따라 조정한 것이므로 기존의 표준 EBP와 개념적으로 일치한다. 또한 이 방법은 그림 2와 3에서 볼 수 있듯이 오류가 커질수록 δ_{pj}값을 크게하고, 오류가 작으면 기존의 표준 EBP와 거의 유사한 δ_{pj}값을 만든다. 이로서 Network Paralysis 현상을 피할 수 있게 된다.

5. 실 험

본 논문에서 제안한 수정된 알고리즘의 성능 평가를 위해 Network Paralysis 현상의 예와 일반적인 문제 해결 예를 기존의 표준 EBP 알고리즘과 비교하는 실험을 실시하였다. 실험에 사용된 신경망의 구조는 3층으로 구성하였으며, 각각 입력, 은닉, 출력 unit의 수가 256×36×18, 35×6×35, 2×2×1, 2×30×2이다. 학습율은 0.1, 0.3, 0.7, 0.9에 대해 각각 실험하였으며 모멘텀은 사용하지 않았다.

실험 1은 Network Paralysis 현상의 예를 문자인식 실험 도중 발견한 예이다. 18개의 16×16 한글 자모음 dot image를 인식하기 위한 예로, 기존의 표준 EBP 알고리즘은 모든 경우에 대해 한 패턴의 한 unit 출력이 오류가 큰 상태에서 (TSS=1.001) 매우 느린 속도로 학습을 진행하고 있었다. 그러나 수정된 알고리즘은 원하는 출력을 매우 빠른 속도로 만들어 냈다. 또한 은닉층 까지 확장시킨 알고리즘의 성능이 더욱 우수함을 보이고 있다.

실험 2는 일반적인 문제 해결 예를 보이기 위한 것으로 encoding problem을 35개의 입력 unit에 대해 35개의 패턴과 10개의 패턴을 갖고 각각 실시하였다.[3] 35개의 패턴을 학습하는 경우는 수정된 알고리즘의 성능이 매우 우수한 것으로 나타났다. 그러나 10개의 패턴에 대한 실험에서는 수정된 알고리즘의 성능이 더 나쁜 것으로 나타났다. 이것은 그림 2에서 빗금친 영역에선 표준 알고리즘에 비해 수정된 알고리즘의 T_S값이 더 작게 되어 생긴 현상으로 추측된다. 그러므로 문제 해결 과정이 단순한 경우, 즉 unit의 출력이 잘못 포화되는 경우가 없는 문제에서는 기존의 표준 알고리즘이 우수하다고 할 수 있다. 그러나 이런 경우 수정된 알고리즘의 T_S의 기울기를 조정하면 다시 우수한 성능을 확보할 수 있다. (eta=0.5일때 괄호안의 숫자가 T_S를 2배로 한 경우 임)

실험 3은 일반 문제 해결 예중 유명한 XOR 문제에 대한 실험을 실시한 결과이다. 출력층에서만 수정한 경우는 실험 2의 두번째 예와 거의 같은 결과를 얻었다. 그러나 은닉층까지 확장하여 수정한 알고리즘은 진동을 일으키며 학습을 하지 못하였다. 이것은 은닉층의 target을 추정치로 대체한 방법에 문제가 있는 것으로 생각된다.

실험 4는 continuous data mapping을 실험하기 위하여 직교좌표를 극좌표로 변환하는 문제에 대

표 2 실험결과
Table 2 Experimental Results

1. 문자인식

input : 18 patterns (cycle)

eta	0.1	0.3	0.5	0.7
표준	11300-(1.02)	45000-(1.001)	9600-(1.003)	924
수정1	5001	2151	1251	964
수정2	2063	733	442	372

('-'는 실험중 강제 정지 된 것으로 이때 tss값이 괄호속의 수임)

2. Encoding problem

input : 35 patterns (cycle)

eta	0.1	0.3	0.5	0.7	0.9
표준	6800-(0.6)	58034	36069(11630)	27503	12202
수정1	35327	11616	6844(3335)	4819	3716
수정2	33043	10996	6622(3408)	4766	3707

input : 10patterns (cycle)

eta	0.1	0.3	0.5	0.7	0.9
표준	6556	2197	1326 (3411)	953	747
수정1	10108	3381	2035 (948)	1457	1135
수정2	9440	3135	1874 (1017)	1332	1031

3. Exclusive OR(XOR) problem

input : 4 patterns (cycle)

eta	0.1	0.3	0.5	0.7	0.9
표준	113176	9835	4854	3227	2003
수정1	67203	12555	5362 (2779)	4702	3178
수정2	oscillation				

4. coordinate conversion function

input : 400 patterns

(cycle)

eta	0.1	0.3	0.5	0.7	0.9
표준	917	339	244	202	169
수정1	506	239	177	134	106
수정2	1190	412	292 (132)	253	225

해 적용한 예이다. 400개의 데이타로 학습시킨 후 무작위 입력을 주어 출력을 확인 하였다. 결과는 수정된 알고리즘이 우수하였지만 은닉층으로의 확장은 실험 2와 같은 결과를 얻었다.

6. 결 론

본 논문에서는 표준 EBP 학습 규칙의 문제점중 하나인 network paralysis현상의 원인을 규명하고 이를 방지하기 위한 수정된 알고리즘을 제안하였다. 이 알고리즘은 표준 EBP 알고리즘의 Activation Function Derivative를 수정한 것으로써 특히 은닉층에까지 확장 적용하였다. 실험 결과 표준 학습 규칙이 학습하기 어려운 문제에 대해 학습을 완수함으로써 새 학습 규칙이 Network Paralysis 현상을 극복할 수 있음을 보였다.

그러나 은닉층으로의 확장은 target을 추정하여 만든것이므로 실험결과로 볼 때 모든 문제에 대해 항상 타당하다고는 할 수 없는 문제가 있다. 또한 이 학습규칙은 Activation Function의 Derivative만을 변경한 것이므로 앞으로 Activation Function자체에 대한 연구가 요구된다.

참 고 문 헌

[1] Ali A. Minai & Ronald D. Williams, "Back-propagation Heuristics : A study of the Extended Delta-Bar-Delta Algorithm," IJCNN, 90, Jun, Vol. 1, pp. 595~600.

[2] Hedong Yang and Clark C. Guest, "Linear Discriminants, Logic Functions, Back-propagation, and Improved Convergence," IJCNN, 90, Jun, Vol. 3, pp. 287~292.

[3] J.L. McClelland and D. E. Rumelhart, "8. Learning Internal Representation by Error Propagation," Parallel Distributed Processing, Vol. 1.

[4] Norio Bada, "A Hybrid Algorithm for Finding the Global Minimum of Error Function of Neural Networks," IJCNN, 90. Jan, Vol. 1, pp. 585~588.

[5] P. Burrascano and P. Lucci, "Smoothing backpropagation cost function by Delta Constraining," IJCNN, 90. Jun, Vol. 3, pp. 75~80

[6] Philip D. Wasserman, Neural Computing : Theory and Practice, Van Nostrand Reinhold, pp. 43~59

[7] Robert Hecht-Nielsen, "Theory of the Back-propagation Neural Network," IJCNN, 89' June, Vol. 1, pp. 593~605

[8] Tariq Samad, "Backpropagation Improvements Based on Heuristic Arguments," IJCNN, 90. Jan. Vol. 1, pp. 565~568